memento

CLAVES PARA AMSTRAD CPC 464 - 664 y 6128

Sistema de base

Daniel Martin



EDICIONES ELISA S.A.

CLAVES PARA AMSTRAD

CPC 464-664 y 6128

Sistema de base

OTRAS OBRAS DEL FONDO EDITORIAL
DE
EDICIONES ELISA, S.A.

DICCIONARIO DEL BASIC, por **D. A. Lien.**
EL DESCUBRIMIENTO DEL COMMODORE 64, por **D.-J. David.**
102 PROGRAMAS PARA ZX81 Y SPECTRUM, por **J. Deconchat.**
102 PROGRAMAS PARA COMMODORE 64, por **J. Deconchat.**
102 PROGRAMAS PARA APPLE II, por **J. Deconchat.**
102 PROGRAMAS PARA EXL 100, por **J. Deconchat.**
CLAVES PARA EL APPLE II, APPLE II PLUS Y APPLE IIe, por **N. Breaud-Pouliquen.**
PASAPORTE PARA APPLESOFT, por **C. Galais.**
EL APPLE Y SUS FICHEROS, por **J. Boisgontier.**
CLAVES PARA EL ZX SPECTRUM, por **J.-F. Séhan.**
CLAVES PARA COMMODORE 64, por **D.-J. David.**
EL BASIC de la A a la Z, por **J. Boisgontier.**
ZX-SPECTRUM PARA TODOS, por **Marcel Henrot y Jacques Boisgontier.**
COMMODORE 64 PARA TODOS, por **J. Boisgontier, S. Brebion** y **G. Foucault.**
CP/M PALABRA POR PALABRA, por **Yvon Dargery.**
EL BASIC Y SUS FICHEROS, por **J. Boisgontier.**
Tomo 1: Métodos prácticos.
Tomo 2: Programas.
DICCIONARIO DEL IBM BASIC. Enciclopedia del lenguaje BASIC para IBM-PC, por **D. A. Lien.**
EXL 100 EN LA ESCUELA, por **D. Nielsen.**

CLAVES PARA AMSTRAD

CPC 464-664 y 6128

Sistema de base

por

Daniel Martin

Versión castellana de

Rafael Guimerà y Francisco Martín



EDICIONES ELISA, S.A. **1986**

Título original de la obra: CLEFS POUR AMSTRAD 1. Système de base.

Primera edición: noviembre 1986.

ISBN: 84-7622-021-9.
Depósito legal: B. 34.369-1986.

Printed in Spain *Impreso en España*

GRAFFING, S.A. — Arquímedes, 18 - HOSPITALET DEL LLOBREGAT

PRESENTACIÓN

El presente libro constituye un auténtico repertorio del Amstrad. Desde la arquitectura interna al Basic, pasando por el lenguaje máquina, el conexionado, la estructura y la programación de los principales circuitos, el soft interno y los trucos, todo queda analizado al mismo tiempo que se ilustra con breves explicaciones y algunos ejemplos.

Tanto el programador-analista como el simple curioso de la informática encontrará en este libro toda la información necesaria para la correcta utilización de su Amstrad.

SUMARIO

ESQUEMA GENERAL Y ARQUITECTURA INTERNA

El diagrama de la página siguiente nos muestra los diferentes circuitos que componen el equipo físico.

El sistema se articula alrededor de una unidad central Z80 con un reloj de 4mhz.

El circuito más importante del Amstrad, si exceptuamos el propio microprocesador, es sin duda el **"GATE ARRAY"**, que contiene toda la lógica de control del sistema. Controla, particularmente, el color, el modo pantalla y gestiona las memorias muertas (ROM).

El "GATE ARRAY" conjuntamente con el CRTC 6845 (Cathode Ray Tube Controller —controlador del tubo de rayos catódicos—) gestiona todas las señales de video del monitor (pantalla).

Otro circuito importante es el PSG AY3-8912 (Programmable Sound Generator —Generador programable de sonido—). Este circuito posee tres canales distintos con un generador de ruido y un controlador de envolvente para cada canal. En el *capítulo V* del presente manual se describirá la manera de programarlo.

El sistema también está provisto de una puerta de entrada-salida que se usa para leer el teclado, y la manecilla de juegos (Joystick).

El último circuito principal es el PPI 8255, que tiene un papel muy importante en la gestión de la manecilla de juegos, en la puerta paralela de la impresora, del magnetófono-cassette y también en la selección de las columnas del teclado.

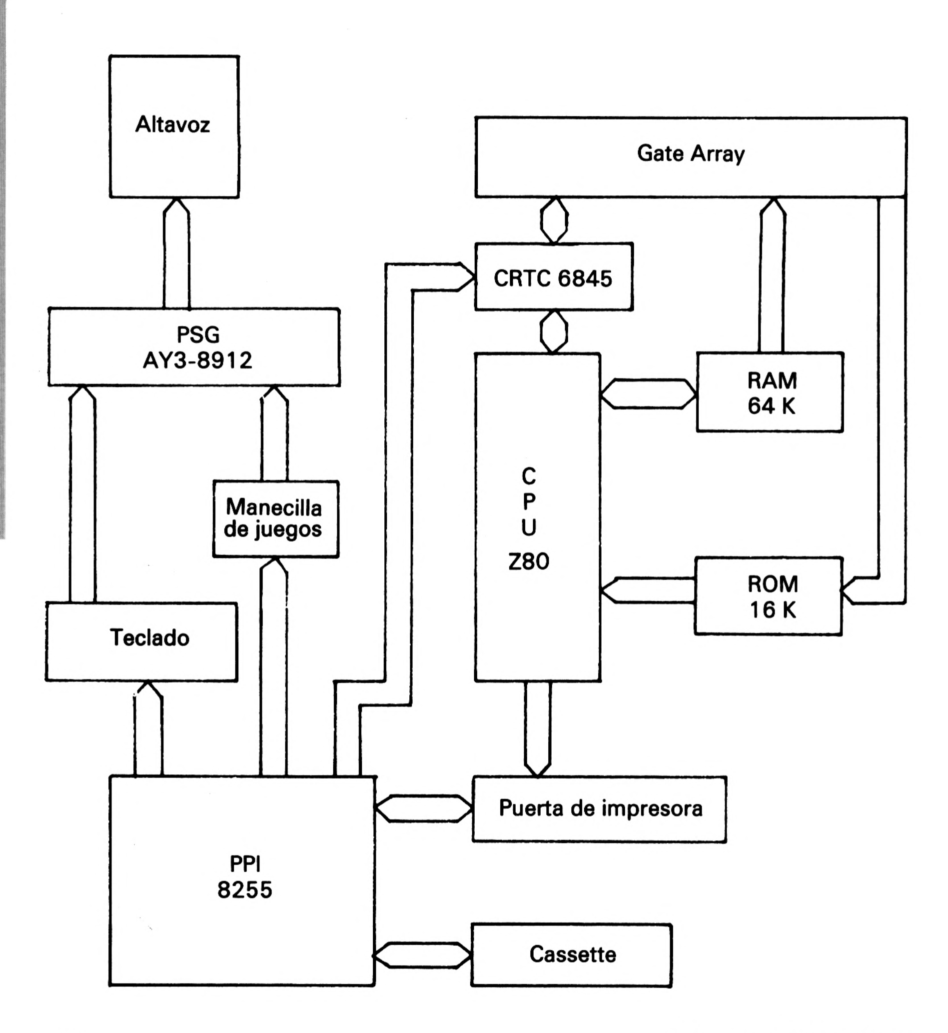
El sistema cuenta con 64 K de memoria viva (RAM) y 32 K de memoria muerta (ROM) que contienen el sistema de explotación del Basic.

La **memoria muerta** (ROM) de 32 K, situada en el circuito central, está dividida lógicamente en dos bloques de 16 K. Los 16 K inferiores van desde la dirección 0000 a la dirección 3FFF, y los 16 superiores ocupan las direcciones C000 a FFFF. Estas dos memorias pueden ser puestas en circuito o suprimidas por el control del "GATE ARRAY".

En la **PUERTA** de extensión encontramos una señal que puede utilizarse para desconectar las memorias muertas internas y permitir que otras memorias externas puedan acceder al microprocesador. Esta posibilidad permite, por ejemplo, instalar un disco flexible.

La **memoria viva** (RAM) está constituida por 64 K bytes dinámicos que van desde la dirección 0000 hasta la FFFF. Los 16 K inferiores y los 16 K superiores se encuentran superpuestos con la ROM. Normalmente esto no representa ningún problema. Cuando se escribe, se hace automáticamente sobre la RAM. Cuando se lee, hay que seleccionar previamente la ROM o la RAM según lo que se quiera leer.

La **memoria de pantalla** se sitúa en la memoria central y ocupa 16 K. Puede hallarse en la dirección 0000, en la dirección 4000, en la dirección 8000 o en la dirección C000. Generalmente, al comienzo, se halla en la dirección C000.



CARACTERÍSTICAS GENERALES

Espacio máximo disponible : 43 533 bytes.

Nombre de las variables : 1 a 40 caracteres.

Datos
Entero : de −32 768 a 32 767.
Simple precisión : de 293 874 E-39 a 170 141 E30 en 9 cifras significativas o en 6 con notación exponencial.
Cadena : 0 a 255 caracteres.

Longitud de una línea de programa : máximo de 255 caracteres.

Número de línea : de 1 a 65 535 ambos inclusive.

Ocupación de memoria : una línea Basic ocupa un mínimo de 6 bytes. Dos para el número de línea, dos para la longitud de la línea, uno para la separación y uno como mínimo para una instrucción (REM, PRINT).

Distribución de variables
Entero positivo de 1 a 9 : 1 byte.
Entero negativo de 1 a 9 : 2 bytes.
Entero positivo de 10 a 255 : 2 bytes.
Entero negativo de 10 a 255 : 3 bytes.
Simple precisión positivo (255-65535) : 3 bytes.
Simple precisión negativo (255-65535) : 4 bytes.
Entero superior a 65535 positivo o no entero positivo : 6 bytes.
Entero superior a 65535 negativo o no entero negativo: 7 bytes.

AFTER

AFTER X,[Y] GOSUB N

Llama un subprograma después de haber esperado X cincuentésimas de segundo. Y es opcional e indica qué cronómetro se utiliza. Hay cuatro, numerados del 0 al 3. Si no se indica el valor de Y se supone igual a 0.

AUTO

AUTO [N],[X]

Numera automáticamente las líneas de un programa empezando por la línea N con un incremento de X. Por omisión es 10 el valor de N y X.

BORDER

BORDER X,[Y]

X e Y representan números de color (de 0 a 26). Permite cambiar el color del borde de la pantalla. Si se indica el valor de Y, los dos colores van alternando a la velocidad que se determine por medio de la orden SPEED INK.

CALL

CALL ADR[,lista de parámetros]

Orden utilizada en Basic para hacer llamar a un subprograma en lenguaje máquina situado en la dirección ADR. Se puede pasar una lista de parámetros a este subprograma.

CAT

CAT

Lee la cassette y proporciona los nombres de los ficheros que encuentra sin borrar el programa que se halla en la memoria central.

CHAIN

CHAIN nombre del programa[,N]

Carga un programa de la cassette en la memoria central suprimiendo el que está en ella en este momento, luego lo ejecuta empezando por la línea número N, o por la línea de número más bajo si no se indica ningún valor para N.

CLEAR

CLEAR

Borra el contenido de todas las variables y de todos los ficheros.

CLG

CLG

Borra la pantalla gráfica.

CLOSEIN

CLOSEIN

Cierra un fichero de cassette abierto en dirección de entrada.

CLOSEOUT

CLOSEOUT

Cierra un fichero de cassette abierto en dirección de salida.

CLS CLS [# n]

Limpia la pantalla o la ventana de la pantalla y la deja del color definido por la última instrucción PAPER. n es el número de canal comprendido entre 0 y 7 que corresponde a la pantalla definida en la instrucción WINDOW.

CONT CONT

Ordena proseguir la ejecución de un programa interrumpido por un STOP, END o *break*, siempre que no se haya modificado el programa.

DATA DATA dato1, dato2, dato3,...

Permite almacenar en un programa la lista de constantes que la función READ leerá.

DEF FN DEF FNf[(X,...)]=expr

Permite definir una función de usuario; f representa el nombre de la función, (X,...) sus parámetros formales y expr su expresión general.

DEFINT DEFINT X-Y DEFINT X,Y,...

Definición de un conjunto de variables en la gama X-Y o siguiendo la lista X,Y,... como si fueran del tipo ENTERO.

DEFREAL DEFREAL X-Y DEFREAL X,Y,...

Definición de un conjunto de variables en la gama X-Y o siguiendo la lista X,Y,... como si fueran del tipo SIMPLE PRECISIÓN.

DEFSTR DEFSTR X-Y DEFSTR X,Y,...

Definición de un conjunto de variables en la gama X-Y o siguiendo la lista X,Y,... como si fueran del tipo ALFANUMÉRICAS. (Cadenas de caracteres.)

DEG DEG

Establece que el cálculo se efectúe en grados. Por omisión, las funciones sinusoides utilizarían radianes para los datos numéricos. La orden DEG se desactiva por las órdenes CLEAR y RAD o al cargar otro programa.

DELETE DELETE (N1,N2,...) DELETE N1- N2

Borra del programa que está en memoria central la serie de líneas (N1,N2,...) o las líneas comprendidas entre la N1 y la N2.

DI Inhibición de las interrupciones. Todas las órdenes que provoquen interrupciones resultarán inoperantes, a excepción de BREAK.

DIM DIM var(n) DIM var(n1,n2,...),var(n1,n2,...)

Dimensión de una tabla (var) de 1 a N dimensiones. Por omisión, una variable queda automáticamente dimensionada a 10 (var(10)).

DRAW DRAW X,Y,A

Dibuja una línea en la pantalla a partir del cursor gráfico y hacia la posición de las coordenadas (X,Y) con el color número A.

DRAWR DRAWR X,Y,A

Dibuja una línea en la pantalla a partir del cursor gráfico hasta la posición relativa (+X,+Y) con respecto al cursor gráfico, con el color número A.

EDIT EDIT N

Pasa la línea número N a modo edición.

EI Habilita las interrupciones. Anula el efecto de la orden DI.

END END

Instrucción de fin de ejecución de un programa.

ENT ENT NE[,SE]

Envolvente de tono. Permite introducir vibratos en una nota: NE representa el número de envolvente (de 0 a 15); SE comprende tres números por sección (número de pasos, altura del paso en frecuencia, tiempo de pausa). Pueden describirse cinco secciones.

ENV ENV NE[,SE]

Envolvente de volumen. Permite definir el tipo de sonido: NE representa el número de envolvente (de 0 a 15); SE comprende tres números por sección (número de pasos, altura en volumen del paso, tiempo para cada paso). Se pueden describir cinco secciones.

ERASE ERASE lista de nombres de variables

Permite liberar el espacio de la memoria reservado por la orden DIM.

ERROR ERROR N

N representa un número entero. Permite detectar un tipo de error determinado y decidir una acción definida cuando ha tenido lugar el error.

EVERY

EVERY N,M GOSUB NL

Significa: cada N centésimas de segundo contadas en el cronómetro número M, ejecutar el subprograma situado en la línea número NL. Hay cuatro cronómetros disponibles y están numerados del 0 al 3. Esta orden permite llamar un subprograma a intervalos regulares.

FOR

FOR var=I TO F [STEP P]

Introduce un bucle. Todas las instrucciones comprendidas entre FOR var=I TO F [STEP P] y la NEXT var correspondiente, se repetirán para todos los valores de var que vayan de I a F de P en P (si no se ha especificado P, de 1 en 1):

```
10 FOR I=1 TO 20 STEP 2
20 PRINT I,"   ",I*I
30 NEXT I
```

GOSUB

GOSUB NL

Llama el subprograma situado en la línea que lleva el número NL.

GOTO

GOTO NL

Efectúa un salto a la línea que lleva el número NL.

IF

IF condición THEN instrucción

Efectúa la instrucción que sigue a THEN si se ha cumplido la condición que sigue a IF.

```
IF A=3 THEN GOSUB 1000
```

INK

INK tinta, color[,color]

Puede haber un cierto número de tintas que sigan al MODO elegido para la pantalla. INK permite cambiar el color de la tinta (tinta) y el de fondo (color). Si se especifican dos colores de fondo, éstos irán alternando cada cincuentésima de segundo.

INPUT

INPUT [# número de canal,][;][cadena;]
lista de variables

Lee los datos provenientes del canal especificado y los asigna a las distintas variables nombradas. El primer [;] suprime el salto de renglón al final del que se escribe. El ";" después de la cadena hace que aparezca en pantalla un ?, mientras que una "," hace que aparezca el mensaje "?redo from start" cuando se teclea algo incorrecto. Cuando se ha especificado un canal de cassette, no aparece el ?
A cada variable de la lista le será asignado un elemento del fichero.

B
A
S
I
C

KEY

KEY número entero, cadena de caracteres

Permite definir una nueva tecla de función. El número entero define la tecla a la que se le asignará la cadena de caracteres. Los valores van de 128 a 140: la tecla 0 del teclado se designa por el número 128, la tecla número 1 por el 129..., la tecla 9 por 137, la tecla "." por 138, la tecla "ENTER" por 139 y las teclas "CTRL" y "ENTER" a la vez por 140.

Ejemplo:

KEY 132,"RUN"+CHR$(13)

Asocia a la tecla 4 del teclado numérico la orden RUN seguida de "ENTER".

KEY DEF

KEY DEF número tecla, repetición, número carácter

Cambia el valor generado por una tecla.
KEY DEF 45,1,65 hace aparecer la A en la tecla de la J, permitiendo la repetición si se mantiene la tecla pulsada.
KEY DEF 46,0,63 hace aparecer el ? en la tecla de la N sin que se permita la repetición.

LET

LET variable= expresión

Asigna a una variable el resultado de la expresión que está a la derecha del signo igual.
LET A=500*3
En Basic AMSTRAD basta con escribir A=500*3.
LET se usa únicamente cuando se quiere que el programa sea compatible con antiguos programas.

LINE INPUT

LINE INPUT [# número de canal,][;]
[cadena;]variable

LINE INPUT "NOMBRE";A$
Lee una línea entera del canal indicado (por omisión es 0). Esta instrucción permite que si en una variable se encuentra una coma, ésta queda incluida en dicha variable, y no como sucedería con un simple INPUT en que la variable quedaría partida por la coma.

LIST

LIST [números de líneas][,# número de canal]

Lista el programa en el canal designado. 0 corresponde a la pantalla y 8 a la impresora. Se puede detener el desplazamiento de la pantalla pulsando una vez la tecla ESC, y proseguir el listado pulsando cualquier tecla. Si se pulsa dos veces la tecla ESC se vuelve al modo directo.

LOAD

LOAD [nombre del fichero][,dirección]

Carga un programa Basic del cassette en memoria central sustitu-

yendo todo lo que se encuentre en ella. En caso de un programa en binario puede especificarse la dirección donde deba cargarse.

LOCATE

LOCATE [# número de canal,]X,Y

Coloca e! cursor de texto en la posición de las coordenadas (X,Y) relativa al origen de la ventana de pantalla (WINDOW). El punto de coordenadas (1,1) es el punto situado en el ángulo superior izquierdo de la ventana.

MEMORY

MEMORY dirección

Permite redefinir la dirección del byte más elevado en el espacio de memoria disponible para el Basic. Por omisión este byte se encuentra en la dirección AB7F.

MERGE

MERGE ["nombre del fichero"]

Es idéntico a LOAD, sólo que no efectúa un NEW antes de cargar el programa. Si hay números de línea idénticos, su contenido será el del nuevo programa. Si no se especifica el nombre del fichero será leído el primer programa que se encuentre. Un programa en cassette precedido del signo "!" está protegido y no podrá ser leído.

MODE

MODE N

Permite cambiar el modo de pantalla (N=0, 1 o 2). Borra la pantalla y se pone en INK0 sea cual sea el PAPER INK utilizado en aquel momento. Cuando se usa esta orden, todas las ventanas de texto y de gráfico vuelven a pantalla entera y los cursores vuelven a su punto de origen.

MOVE

MOVE X,Y

Coloca el cursor gráfico en la posición absoluta de coordenadas (X,Y).

MOVER

MOVER X,Y

Coloca el cursor gráfico en una posición de coordenadas (X,Y) relativa a la posición actual.

NEW

Limpia la memoria.
Desaparecen los programas y variables, pero permanecen las definiciones de las teclas (KEY) y la visualización en pantalla.

NEXT

FOR I=1 TO 10 : NEXT [I]

Determina el fin de un bucle iniciado por FOR.

ON...GOTO

ON n GOTO lista de números de líneas

ON...GOSUB ON n GOSUB lista de números de líneas
Para ON A GOTO 100,110,130,132,170,300,320,1000:
si A=1, la bifurcación se hará a la línea 100
si A=2, la bifurcación se hará a la línea 110
.. ..
si A=7, la bifurcación se hará a la línea 320,...

ON BREAK GOSUB ON BREAK GOSUB número de línea
Llama un subprograma cuando se da un BREAK en medio de un programa. Se efectúa un BREAK pulsando dos veces consecutivas la tecla ESC.

ON BREAK STOP Anula el efecto de la instrucción ON BREAK GOSUB.

ON ERROR GOTO ON ERROR GOTO número de línea
Bifurca la ejecución del programa a la línea indicada cuando se produce un error.

ON SQ GOSUB ON SQ (n) GOSUB nl
Ejecuta la subrutina situada en la línea nl cuando la cola sonora que corresponde al canal sonoro (n) no está llena. (n) sólo puede valer 1, 2 o 4 correspondiéndose con los canales A, B o C.

OPENIN OPENIN "nombre del fichero"
Abre un fichero cassette para que el programa que está en memoria central pueda utilizar los datos que se encuentran en dicho fichero. Si el nombre del fichero está precedido de un signo "!", no aparecen los mensajes habituales de manejo del cassette, y el programa lee directamente el primer bloque del fichero.

OPENOUT OPENOUT "nombre del fichero"
Abre un fichero en el cassette para que el programa pueda ir a él y alojar allí los datos. Si el nombre del programa está precedido del signo "!", no aparecen los mensajes habituales de manejo del cassette. El programa crea, inmediatamente, el primer bloque de transferencia (buffer) de 2 Kbytes, y no se escribe nada en el cassette hasta tanto que el buffer no esté lleno o hasta que una orden CLOSE no cierre el fichero.

ORIGIN ORIGIN X,Y [,I,D,A,O]
Determina las coordenadas (X,Y) del punto de partida del cursor gráfico. Los elementos opcionales I,D,A,O permiten definir una nueva ventana.

OUT OUT número de puerta, número entero
Envía el número entero a la puerta especificada. El número entero puede estar comprendido entre 0 y 255 y el número de puerta entre 0 y 65535.

PAPER PAPER [# número de canal,] número de tinta
Define el color de fondo de los próximos caracteres que se escribirán en la pantalla.

PEN PEN [# número de canal,] número de tinta
Define el color de los próximos caracteres que se escribirán en la pantalla.

PLOT PLOT X,Y [, número de tinta]
Visualiza en la pantalla el punto de coordenadas (X,Y) en el color precisado por el número de tinta o, por omisión, en el color utilizado anteriormente.

PLOTR PLOTR X,Y [, número de tinta]
Visualiza en pantalla el punto de coordenadas (X,Y) relativas a la posición del cursor. Se puede elegir el color.

POKE POKE dirección, dato
Inscribe el dato en la dirección precisada.

PRINT PRINT [# número de canal,] datos
Imprime los datos en el canal precisado
(0 = pantalla, por omisión).
PRINT USING permite especificar diversos formatos de edición.

RAD Establece el modo RADIANES.

RANDOMIZE RANDOMIZE [N]
Fija la secuencia de números seudoaleatorios a partir de N, siendo N un número entero comprendido entre 0 y 65535. Por omisión N = 0.

READ READ lista de variables
Lee los datos contenidos en las líneas de DATA y los asigna a las variables precisadas.

RELEASE RELEASE canales sonoros
Permite liberar un canal sonoro en espera.

REM

Introduce una línea de comentarios.

RENUM

RENUM [nnl,][anl,][d]

Renumera las líneas del programa presente en la memoria central. nnl = nuevo número de línea, que por omisión vale 10. anl = antiguo número de línea, que por omisión es el número de la primera línea del programa. d = diferencia entre dos líneas, que por omisión vale 10.

RESTORE

RESTORE [número de línea]

Define la línea en la que debe comenzar la lectura de los DATAs. Si no se especifica ningún número de línea, la lectura empieza en la primera línea que contenga DATAs.

RESUME

RESUME [número de línea de recomienzo]

Permite continuar la ejecución de un programa a partir de la línea de recomienzo, después que se ha detectado un error y se ha corregido por una orden ON ERROR GOTO.

RETURN

Retorna al programa principal después de una llamada a un subprograma, por medio de una instrucción GOSUB...

RUN

RUN [número de línea]

Ejecuta el programa presente en la memoria central partiendo del número de línea que se especifica o, por omisión, partiendo del número de línea más bajo.

RUN "nombre del programa"

Carga el programa especificado que está en el cassette e inicia su ejecución. Si no se pone el nombre del programa (RUN""), el Basic carga y ejecuta el primer programa que encuentra en el cassette.

SOUND

SOUND canal,periodo [,duración[,volumen[,envolvente de volumen[,envolvente de tono]]]]

Produce el sonido especificado.

Canal: se pueden seleccionar los tres canales A, B y C a la vez, de dos en dos o por separado. Para una mejor comprensión se recomienda leer las explicaciones referentes al CHIP AY3-8912.

Periodo: toma los valores comprendidos entre 0 y 4095. La frecuencia del sonido se obtiene dividiendo 125 000 por el periodo.

Duración: puede tomar los valores comprendidos entre −32768 y +32767. Por omisión toma el valor de 20. Si la duración tiene valor positivo, representa un número de centenas de segundos; si tiene valor negativo, representa un número de repeticiones de envolvente de volumen completas.

Volumen: toma un valor comprendido entre 0 y 15. Por omisión, toma el valor de 15, si se ha formulado una orden ENV, y si no el valor de 4.

Envolvente de volumen: puede tomar un valor comprendido entre 0 y 15 (0 por omisión) e indica el tipo de envolvente definida en la instrucción ENV.

Envolvente de tono: puede tomar un valor comprendido entre 0 y 15 (0 por omisión) e indica el tipo de envolvente de tono definida por la orden ENT.

SPEED INK

SPEED INK número entero, número entero

Permite modificar la velocidad de alternancia de los colores de fondo en el caso de que se hayan definido dos colores con la orden INK:

```
10 INK 0,1,9
20 SPEED INK 100,20
```

SPEED KEY

SPEED KEY espera, periodo de repetición

Permite regular el tiempo durante el que debe pulsarse una tecla para que se produzca repetición (espera), y la velocidad de repetición (periodo de repetición). Estas regulaciones se hacen cada cincuentavo de segundo y con valores de 10,10 por omisión.

SPEED WRITE

SPEED WRITE n

n = 1 o 0

Permite modificar la velocidad de grabación de un programa en el cassette. En el momento de la lectura (LOAD), el CPC464 establece automáticamente la velocidad correcta de lectura.

Para n = 0, la velocidad de lectura será de 1000 baudios; para n = 1, la velocidad será de 2000 baudios. Por omisión n = 0.

STOP

Permite detener la ejecución de un programa al mismo tiempo que deja al operador la posibilidad de continuarla por medio de la orden CONT.

SYMBOL

SYMBOL número de carácter, lista de caracteres

Permite redefinir el carácter cuyo número se indica. Se pueden redefinir todos los caracteres comprendidos entre 240 y 255. Si se desea redefinir otros caracteres véase la orden SYMBOL AFTER.

SYMBOL AFTER

SYMBOL AFTER número entero

Fija el número de caracteres que se desea redefinir. Por omisión vale 240.

TAG

TAG [# número de canal]

Permite escribir caracteres en la posición del cursor gráfico. De este modo se pueden mezclar textos con gráficos.

```
10 MOVE 200,300
20 PRINT "AQUÍ"
30 TAG
40 PRINT "HOLA"
```

"AQUÍ" quedará escrito en la posición del cursor de texto mientras que "HOLA" quedará escrito en la posición del cursor gráfico (200,300).

TAGOFF

TAGOFF [# número de canal]

Anula los efectos de la orden TAG en el canal especificado (0 por omisión) y reenvía el texto al lugar donde estaba el cursor antes de la orden TAG.

TRON

Validación del modo TRACE

TROFF

Invalidación del modo TRACE

En el modo TRACE, aparecen en la pantalla todos los números de las líneas por las que va pasando el programa durante su ejecución. Este modo es muy útil durante la elaboración de un programa.

WAIT

WAIT nPUERTA, byte de máscara, byte de selección

Permite esperar que una determinada combinación de bytes, presente en una puerta, adquiera un valor específico. Esta instrucción lee el contenido de la puerta nPUERTA, le aplica la función AND LÓGICA con el byte de máscara, después una función OR EXCLUSIVO con el byte de selección y sólo devuelve el control al programa si el resultado es distinto de 0. La función de máscara permite aislar el o los bit(s) que hay que comprobar. La función de selección permite invertir el estado que hay que comprobar.

WEND

Termina el bucle iniciado por la orden WHILE.

WHILE

WHILE expresión lógica

WHILE X>2 ejecutará las líneas de programa entre WHILE y WEND mientras la expresión lógica sea verdadera (aquí, mientras X sea mayor que 2).

El siguiente programa:

```
10 X=4:Y=0
20 WHILE X<>Y
30 INPUT "cuánto son 2 y 2 ";Y
40 WEND
50 PRINT "muy bien":END
```

hará exactamente lo mismo que el programa:

```
10 X=4
20 INPUT "cuánto son 2 y 2 ";Y
30 IF X<>Y THEN GOTO 20
40 PRINT "muy bien":END
```

La utilidad de las instrucciones WHILE y WEND sólo se evidencia si se practica la programación estructurada. En efecto, uno de los principios de este tipo de programación es evitar sistemáticamente las instrucciones de bifurcación (GOTO,...) para que los programas sean más legibles.

WIDTH

WIDTH número entero

Indica al Basic la longitud de las líneas de la impresora en número de caracteres.

WINDOW

WINDOW [# número de canal,] izquierda, derecha, arriba, abajo

Permite definir una ventana de texto para un canal dado de pantalla. Los canales del 0 al 7 pueden utilizarse para definir las ventanas de texto de la pantalla.

WINDOW SWAP

WINDOW SWAP número de canal, número de canal

Permite intercambiar los contenidos de las dos ventanas.

WRITE

WRITE [# número de canal,][lista de datos]

Escribe en el canal indicado (0 por omisión) la lista de datos que se quiera sin cambiar la puntuación.

WRITE "AQUÍ", 23, 5 escribirá en la pantalla

"AQUÍ", 23, 5

XPOS

Proporciona la abscisa (posición horizontal) del cursor gráfico.

YPOS

Proporciona la ordenada (posición vertical) del cursor gráfico.

ZONE

ZONE, número entero

Permite modificar el espacio comprendido entre dos tabulaciones automáticas obtenidas por una coma. Esta tabulación por omisión vale 13.

```
ZONE 2:PRINT 1,2,3
```

ABS	ABS (expresión numérica) Proporciona el valor absoluto de la expresión numérica.
ASC	ASC (cadena de caracteres) Proporciona el código ASCII del primer carácter contenido en la cadena de caracteres.
ATN	ATN (expresión numérica) Proporciona en radianes o en grados el valor del ángulo cuya tangente vale la expresión numérica.
BIN$	BIN$ (número entero decimal [,N]) Convierte el número entero decimal en un número binario expresado en N caracteres (8 por omisión).
CHR$	CHR$ (N) Proporciona el carácter cuyo código ASCII es N. N es un número entero comprendido entre 0 y 255.
CINT	CINT (expresión numérica) Convierte una expresión numérica en un número entero redondeando a la unidad superior cuando la parte decimal de la expresión es igual o superior a 0,5.
COS	COS (valor del ángulo) Proporciona el valor del coseno de un ángulo que se supone expresado en radianes. Puede expresarse en grados escribiendo de antemano la orden DEG.
CREAL	CREAL (expresión numérica) Convierte un número entero en un número real. Es la función inversa de la función CINT.
EOF	PRINT EOF Indicador de fin de fichero de cassette. Toma el valor −1 cuando la entrada de cassette encuentra el fin de fichero, y el valor 0 en todos los demás casos.
ERR	Variable que contiene el número del último error que se ha producido.
ERL	Variable que contiene el número de la línea donde se ha producido el último error.

B
A
S
I
C

EXP

PRINT EXP (n)

Calcula e elevado a n (exponencial).

FIX

FIX (n)

Suprime la parte decimal del número n sin redondear al entero más próximo.

FRE

FRE (X)   FRE (" ")

Proporciona el número de bytes que quedan libres en memoria.

HEX$

HEX$ (n)

Convierte el número entero n en hexadecimal.

HIMEM

Proporciona la dirección más alta utilizable por el Basic.

INKEY

INKEY (N)

Rastrea el teclado para ver qué tecla ha sido pulsada. Si se ha pulsado la tecla que lleva el número N, INKEY (N) vale 0; si se ha pulsado la tecla que lleva el número N al mismo tiempo que la tecla SHIFT, INKEY (N) vale 32; finalmente si no se ha pulsado ninguna tecla o cualquier otra distinta, entonces INKEY (N) vale −1.

```
 5 CLS
10 IF INKEY(54)=32 THEN 30 ELSE IF INKEY(54)=0
   THEN 40
20 GOTO 10
30 PRINT "Se ha pulsado SHIFT y B":GOTO 50
40 PRINT "Se ha pulsado B"
50 GOTO 10
```

INKEY$

A$=INKEY$

Asigna a la variable alfanumérica A$ el valor de la tecla que se acaba de pulsar en el teclado.

Esta función es muy útil para esperar una respuesta determinada sin tener que pulsar la tecla RC.

```
10 CLS
20 PRINT "Toma usted azúcar en el café ?";
30 A$=INKEY$:IF A$="" THEN 30
40 IF A$ <>"0" AND A$<>"N" THEN 30
50 PRINT A$
```

INP

PRINT INP (número de puerta de entrada/salida)

Lee el contenido de la puerta de entrada/salida especificada.

INSTR

INSTR ([N,]A$,B$)

Si la cadena B$ es un fragmento de A$, INSTR(A$,B$) toma un valor numérico igual al número del carácter de A$ donde empieza la cadena B$.

PRINT INSTR ("INÚTIL","UT") dará: 2.

Si se indica N, la comparación empezará a partir del eNésimo carácter de la cadena A$.

INT

INT (expresión numérica)

Suprime la parte decimal y redondea al número entero menor. Es idéntica a FIX para los números positivos, pero para los números negativos que no sean enteros dará una unidad menos que para FIX.

JOY

JOY (N)

Lee el estado de la manecilla de juegos (Joystick). N indica el número de la manecilla (0 o 1).

El resultado de la función se expresa en 6 bits. Si la manecilla está en reposo, los 6 bits valen 0. Los bits pasan a valer 1 en función de la posición de la manecilla o de la presión sobre los botones de tiro.

bit 0 = 1 : manecilla hacia arriba	:	valor = 1
bit 1 = 1 : manecilla hacia abajo	:	valor = 2
bit 2 = 1 : manecilla hacia izquierda	:	valor = 4
bit 3 = 1 : manecilla hacia derecha	:	valor = 8
bit 4 = 1 : botón de tiro 1 pulsado	:	valor = 16
bit 5 = 1 : botón de tiro 2 pulsado	:	valor = 32

Es posible la combinación de varias acciones.
Ejemplo: Si se desplaza la manecilla hacia abajo, a la derecha y se pulsa el botón de tiro 1, la función JOY proporcionará un valor igual a la suma de los valores que proporcionaría cada acción por separado:

desplazamiento hacia abajo	:	2
desplazamiento hacia la derecha	:	8
botón de tiro 1 pulsado	:	16
valor proporcionado	:	2+8+16=26

LEFT$

LEFT$ (cadena,N)

Extrae los N primeros caracteres a la izquierda de la cadena indicada, siendo N un número entero.
PRINT LEFT$ ("AMSTRAD",4) dará: AMST.

LEN

LEN (cadena)

Determina la longitud de una cadena de caracteres, es decir el número de caracteres que la constituyen.

B A S I C

LOG LOG (X)
Calcula el logaritmo en base e de X.

LOG10 LOG10 (X)
Calcula el logaritmo en base 10 de X.

**LOWER$** LOWER$ (cadena)
Transforma todas las mayúsculas en minúsculas de la cadena alfanumérica indicada.

MAX MAX (lista de expresiones numéricas)
Proporciona el máximo valor contenido en la lista de expresiones numéricas.
PRINT MAX (2,67,34,987,12,9,876,0) dará 987.

**MID$** MID$ (cadena,N[,M])
Extrae M caracteres de la cadena empezando por el eNésimo carácter. M vale 1 por omisión.

MIN MIN (lista de expresiones numéricas)
Proporciona el mínimo valor contenido en la lista de expresiones numéricas.

PEEK PEEK (n)
Lectura del valor contenido en la dirección de memoria n.

PI PRINT PI da 3.14159265
Da el valor aproximado del número PI.

POS POS (# número de canal)
Indica la posición horizontal actual del cursor de texto por un canal dado (coordenada X).
En el caso de que el canal indicado sea el de la impresora, POS da la posición del carro, que es 1 si está al margen izquierdo.

REMAIN REMAIN (N)
Suprime el cronómetro indicado (N=0, 1, 2 o 3) y lee el tiempo que llevaba. Indica 0 si no se había puesto en marcha el cronómetro.

**RIGHT$** RIGHT$ (cadena,N)
Extrae N caracteres por la derecha de la cadena de caracteres indicada.

RND	RND (N) Proporciona un número seudoaleatorio, el eNésimo de una secuencia determinada por la orden RANDOMIZE.
ROUND	ROUND (expresión numérica [,N]) Redondea la expresión numérica a N cifras después de la coma. N es un número entero, que por omisión vale 0.
SGN	SGN (expresión numérica) Determina el signo de la expresión numérica. Da −1 si es negativa, 0 si es nula, y 1 si es positiva.
SIN	SIN (valor del ángulo) Proporciona el valor del seno de un ángulo que se supone expresado en radianes. Se puede elegir expresarlo en grados escribiendo de antemano la orden DEG.
SPACE$	SPACE$ (N) Crea una cadena de N espacios blancos, siendo N un número entero.
SQ	SQ (canal sonoro) Determina el número de sitios libres en una cola de sonido para un canal dado.
SQR	SQR (N) Calcula la raíz cuadrada del número N.
STR$	STR$ ([&]N) Convierte la expresión numérica N en una cadena de caracteres. Si la expresión numérica va precedida del signo &, es considerada como un número hexadecimal y se pasará a número decimal antes de ser convertida en cadena de caracteres. PRINT STR$(123) dará 123 en forma de cadena alfanumérica. PRINT STR$(&10) dará 16 en forma de cadena alfanumérica.
STRING$	STRING$ (N,caracteres) PRINT STRING$ (4,"*") dará **** Crea una cadena de caracteres formada por N veces el carácter indicado. N puede ser expresado en hexadecimal con tal de que le preceda el signo &.

TAN

TAN (valor del ángulo)

Proporciona el valor de la tangente de un ángulo que se supone expresado en radianes. Se puede elegir expresarlo en grados escribiendo de antemano la orden DEG.

TEST

TEST (x,y)

Proporciona el valor de la tinta utilizada en el lugar de las coordenadas absolutas (x,y) de la pantalla.

TESTR

TESTR (x,y)

Proporciona el valor de la tinta utilizada en el lugar de las coordenadas relativas (x,y) a la posición actual del cursor en la pantalla.

TIME

Proporciona el tiempo transcurrido en 1/300 de segundo desde la puesta en marcha del ordenador, sin tener en cuenta los tiempos de lectura y escritura en el cassette.

UNT

UNT (número)

Convierte un número entero sin signo en un entero comprendido entre −32767 y 32768.

PRINT UNT(&7FFF) y PRINT UNT(32767)	=	32767
PRINT UNT(&0010) y PRINT UNT(16)	=	16
PRINT UNT(&0001) y PRINT UNT(1)	=	1
PRINT UNT(&FFFF) y PRINT UNT(65535)	=	−1
PRINT UNT(&FFF6) y PRINT UNT(65526)	=	−10
PRINT UNT(&8000) y PRINT UNT(32768)	=	−32768

UPPER$

UPPER$ (cadena)

Transforma en mayúsculas las minúsculas de una cadena.

VAL

VAL (cadena)

Transforma una cadena en una expresión numérica. Dará 0 si la cadena empieza por una letra.

PRINT VAL("34E") dará 34
PRINT VAL("123") dará 123
PRINT VAL("A34") dará 0

VPOS

VPOS (# número de canal)

Proporciona la posición vertical del cursor de texto para el canal indicado (coordenada Y).

XPOS

Proporciona la posición horizontal del cursor gráfico.

YPOS

Proporciona la posición vertical del cursor gráfico.

Todos los códigos inferiores a 127 van precedidos de un byte a 255 (FF).

Cód. Dec	*Cód. Hex*	*Pal- cla*	*Cód. Dec*	*Cód. Hex*	*Pal- cla*
255+ 0	FF+ 0	**ABS**	255+ 27	FF+1B	**UNT**
255+ 1	FF+ 1	**ASC**	255+ 28	FF+1C	**UPPER$**
255+ 2	FF+ 2	**ATN**	255+ 29	FF+1D	**VAL**
255+ 3	FF+ 3	**CHR$**	255+ 64	FF+40	**EOF**
255+ 4	FF+ 4	**CINT**	255+ 65	FF+41	**ERR**
255+ 5	FF+ 5	**COS**	255+ 66	FF+42	**HIMEM**
255+ 6	FF+ 6	**CREAL**	255+ 67	FF+43	**INKEY$**
255+ 7	FF+ 7	**EXP**	255+ 68	FF+44	**PI**
255+ 8	FF+ 8	**FIX**	255+ 69	FF+45	**RND**
255+ 9	FF+ 9	**FRE**	255+ 70	FF+46	**TIME**
255+ 10	FF+ A	**INKEY**	255+ 71	FF+47	**XPOS**
255+ 11	FF+ B	**INP**	255+ 72	FF+48	**YPOS**
255+ 12	FF+ C	**INT**	255+113	FF+71	**BIN$**
255+ 13	FF+ D	**JOY**	255+114	FF+72	**DEC$**
255+ 14	FF+ E	**LEN**	255+115	FF+73	**HEX$**
255+ 15	FF+ F	**LOG**	255+116	FF+74	**INSTR**
255+ 16	FF+10	**LOG10**	255+117	FF+75	**LEFT$**
255+ 17	FF+11	**LOWER$**	255+118	FF+76	**MAX**
255+ 18	FF+12	**PEEK**	255+119	FF+77	**MIN**
255+ 19	FF+13	**REMAIN**	255+120	FF+78	**POS**
255+ 20	FF+14	**SGN**	255+121	FF+79	**RIGHT$**
255+ 21	FF+15	**SIN**	255+122	FF+7A	**ROUND**
255+ 22	FF+16	**SPACE$**	255+123	FF+7B	**STRING$**
255+ 23	FF+17	**SQ**	255+124	FF+7C	**TEST**
255+ 24	FF+18	**SQR**	255+125	FF+7D	**TESTR**
255+ 25	FF+19	**STR$**	255+127	FF+7F	**VPOS**
255+ 26	FF+1A	**TAN**			

Los códigos siguientes no van precedidos de 255.

Cód. Dec	*Cód. Hex*	*Pal- cla*	*Cód. Dec*	*Cód. Hex*	*Pal- cla*
128	80	**AFTER**	138	8A	**CLS**
129	81	**AUTO**	139	8B	**CONT**
130	82	**BORDER**	140	8C	**DATA**
131	83	**CALL**	141	8D	**DEF**
132	84	**CAT**	142	8E	**DEFINT**
133	85	**CHAIN**	143	8F	**DEFREAL**
134	86	**CLEAR**	144	90	**DEFSTR**
135	87	**CLG**	145	91	**DEG**
136	88	**CLOSEIN**	146	92	**DELETE**
137	89	**CLOSEOUT**	147	93	**DIM**

BASIC

Cód. Dec	*Cód. Hex*	*Pal- cla*	*Cód. Dec*	*Cód. Hex*	*Pal- cla*
148	94	**DRAW**	195	C3	**READ**
149	95	**DRAWR**	196	C4	**RELEASE**
150	96	**EDIT**	197	C5	**REM**
151	97	**ELSE**	198	C6	**RENUM**
152	98	**END**	199	C7	**RESTORE**
153	99	**ENT**	200	C8	**RESUME**
154	9A	**ENV**	201	C9	**RETURN**
155	9B	**ERASE**	202	CA	**RUN**
156	9C	**ERROR**	203	CB	**SAVE**
157	9D	**EVERY**	204	CC	**SOUND**
158	9E	**FOR**	205	CD	**SPEED**
159	9F	**GOSUB**	206	CE	**STOP**
160	A0	**GOTO**	207	CF	**SYMBOL**
161	A1	**IF**	208	D0	**TAG**
162	A2	**INK**	209	D1	**TAGOFF**
163	A3	**INPUT**	210	D2	**TROFF**
164	A4	**KEY**	211	D3	**TRON**
165	A5	**LET**	212	D4	**WAIT**
166	A6	**LINE**	213	D5	**WEND**
167	A7	**LIST**	214	D6	**WHILE**
168	A8	**LOAD**	215	D7	**WIDTH**
169	A9	**LOCATE**	216	D8	**WINDOW**
170	AA	**MEMORY**	217	D9	**WRITE**
171	AB	**MERGE**	218	DA	**ZONE**
172	AC	**MID$**	219	DB	**DI**
173	AD	**MODE**	220	DC	**EI**
174	AE	**MOVE**	234	EA	**TAB**
175	AF	**MOVER**	235	EB	**THEN**
176	B0	**NEXT**	236	EC	**TO**
177	B1	**NEW**	237	ED	**USING**
178	B2	**ON**	238	EE	**>**
179	B3	**ON BREAK**	239	EF	**=**
180	B4	**ON ERROR GOTO**	240	F0	**>=**
181	B5	**ON SQ**	241	F1	**<**
182	B6	**OPENIN**	242	F2	**< >**
183	B7	**OPENOUT**	243	F3	**<=**
184	B8	**ORIGIN**	244	F4	**+**
185	B9	**OUT**	245	F5	**-**
186	BA	**PAPER**	246	F6	*****
187	BB	**PEN**	247	F7	**/**
188	BC	**PLOT**	248	F8	
189	BD	**PLOTR**	249	F9	****
190	BE	**POKE**	250	FA	**AND**
191	BF	**PRINT**	251	FB	**MOD**
192	C0	**'**	252	FC	**OR**
193	C1	**RAD**	253	FD	**XOR**
194	C2	**RANDOMIZE**	254	FE	**NOT**

Carácter	Código ASCII		
	Hexadecimal	Decimal	Octal
NUL (CTRL @)	00	0	000
SOH (CTRL A)	01	1	001
STX (CTRL B)	02	2	002
ETX (CTRL C)	03	3	003
EOT (CTRL D)	04	4	004
ENQ (CTRL E)	05	5	005
ACK (CTRL F)	06	6	006
BEL (CTRL G)	07	7	007
BS (CTRL H)	08	8	010
HT (CTRL I)	09	9	011
LF (CTRL J)	0A	10	012
VT (CTRL K)	0B	11	013
FF (CTRL L)	0C	12	014
CR (CTRL M)	0D	13	015
SO (CTRL N)	0E	14	016
SI (CTRL O)	0F	15	017
DLE (CTRL P)	10	16	020
DC1 (CTRL Q)	11	17	021
DC2 (CTRL R)	12	18	022
DC3 (CTRL S)	13	19	023
DC4 (CTRL T)	14	20	024
NAK (CTRL U)	15	21	025
SYN (CTRL V)	16	22	026
ETB (CTRL W)	17	23	027
CAN (CTRL X)	18	24	030
EM (CTRL Y)	19	25	031
SUB (CTRL Z)	1A	26	032
ESC	1B	27	033
FS	1C	28	034
GS	1D	29	035
RS	1E	30	036
US	1F	31	037
SP	20	32	040
!	21	33	041
"	22	34	042
#	23	35	043
$	24	36	044
%	25	37	045
&	26	38	046
'	27	39	047
(	28	40	050
)	29	41	051
*	2A	42	052
+	2B	43	053
,	2C	44	054
-	2D	45	055

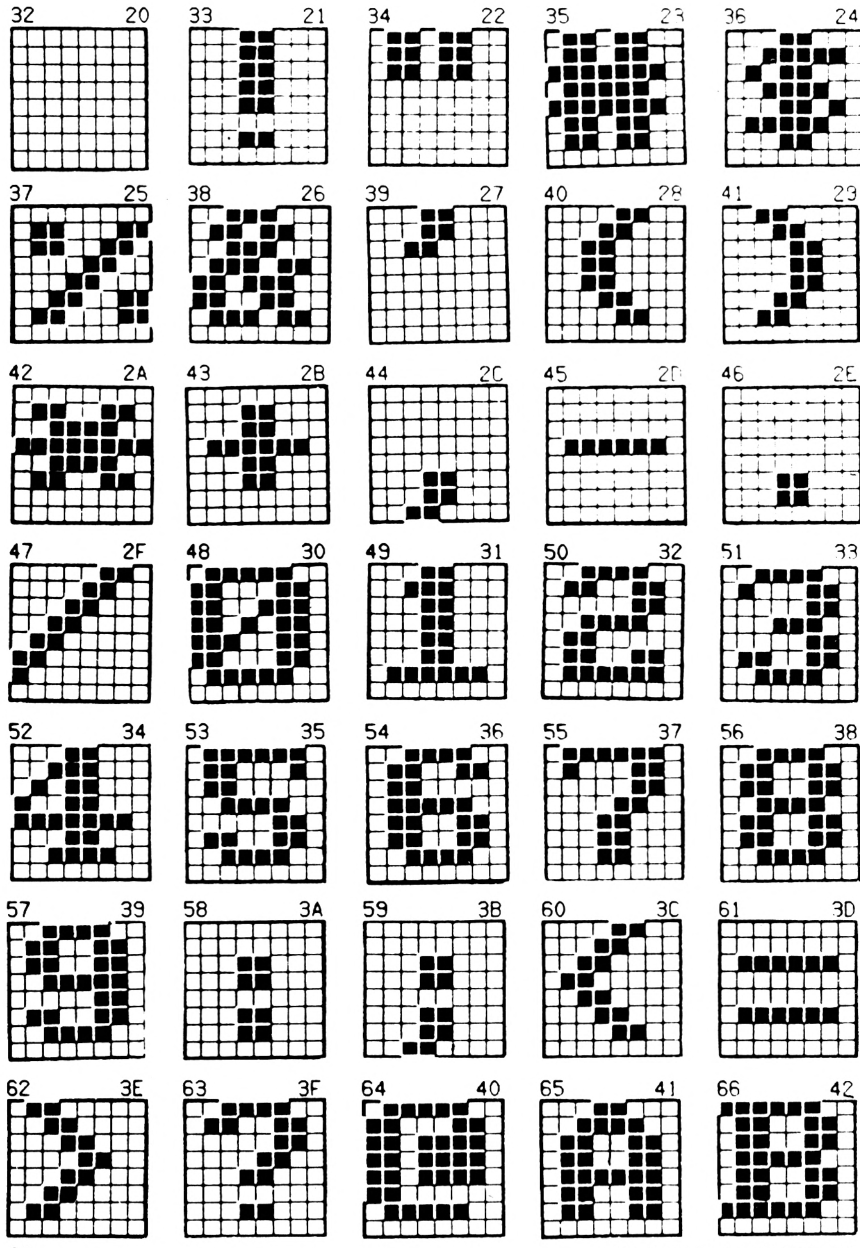
BASIC

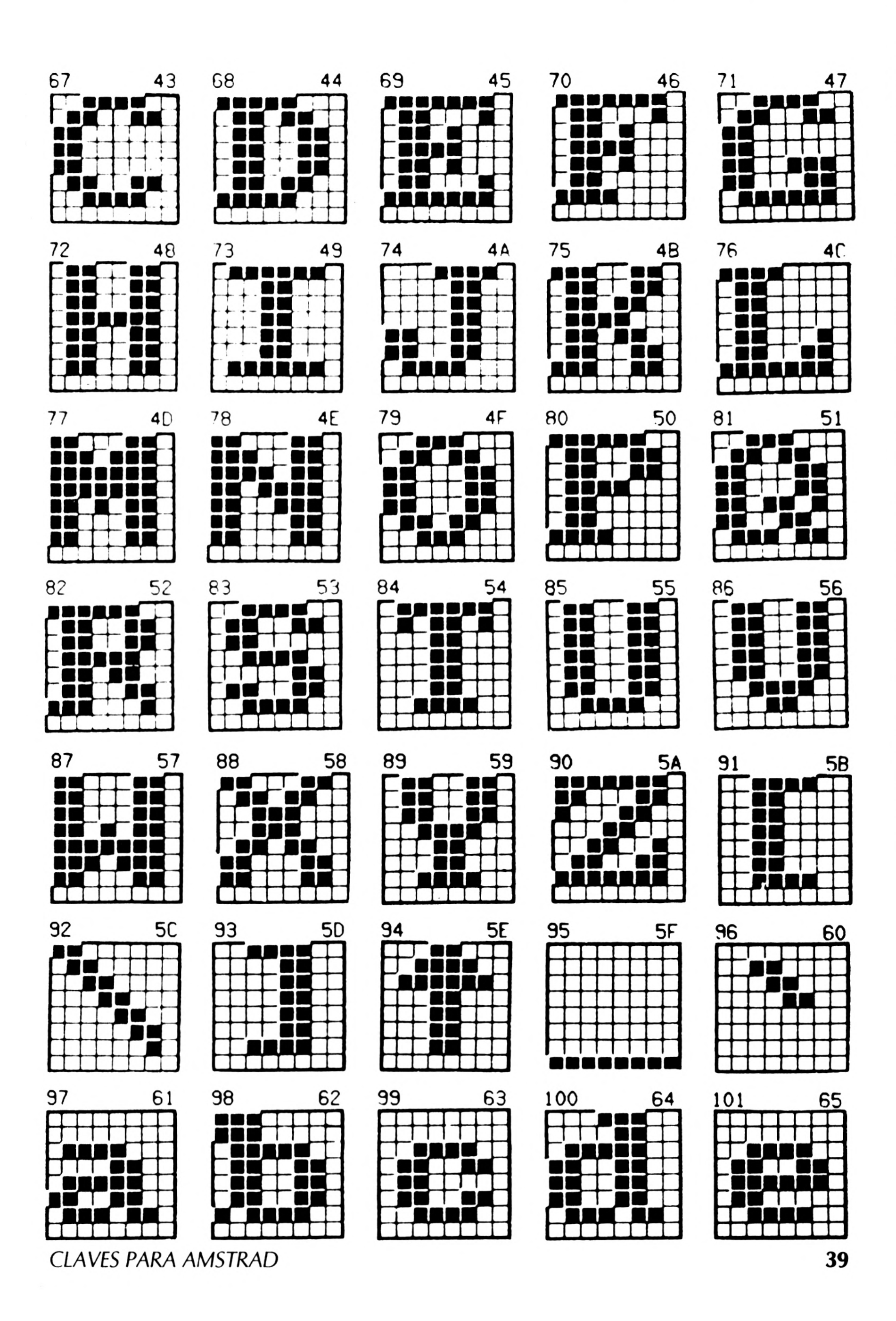
Carácter	*Código ASCII*		
	Hexadecimal	*Decimal*	*Octal*
.	2E	46	056
/	2F	47	057
0	30	48	060
1	31	49	061
2	32	50	062
3	33	51	063
4	34	52	064
5	35	53	065
6	36	54	066
7	37	55	067
8	38	56	070
9	39	57	071
:	3A	58	072
;	3B	59	073
<	3C	60	074
=	3D	61	075
>	3E	62	076
?	3F	63	077
@	40	64	100
A	41	65	101
B	42	66	102
C	43	67	103
D	44	68	104
E	45	69	105
F	46	70	106
G	47	71	107
H	48	72	110
I	49	73	111
J	4A	74	112
K	4B	75	113
L	4C	76	114
M	4D	77	115
N	4E	78	116
O	4F	79	117
P	50	80	120
Q	51	81	121
R	52	82	122
S	53	83	123
T	54	84	124
U	55	85	125
V	56	86	126
W	57	87	127
X	58	88	130
Y	59	89	131
Z	5A	90	132
[	5B	91	133

Carácter	*Código ASCII*		
	Hexadecimal	*Decimal*	*Octal*
\	5C	92	134
]	5D	93	135
↑	5E	94	136
	5F	95	137
è	60	96	140
a	61	97	141
b	62	98	142
c	63	99	143
d	64	100	144
e	65	101	145
f	66	102	146
g	67	103	147
h	68	104	150
i	69	105	151
j	6A	106	152
k	6B	107	153
l	6C	108	154
m	6D	109	155
n	6E	110	156
o	6F	111	157
p	70	112	160
q	71	113	161
r	72	114	162
s	73	115	163
t	74	116	164
u	75	117	165
v	76	118	166
w	77	119	167
x	78	120	170
y	79	121	171
z	7A	122	172
{	7B	123	173
|	7C	124	174
}	7D	125	175
~	7E	126	176

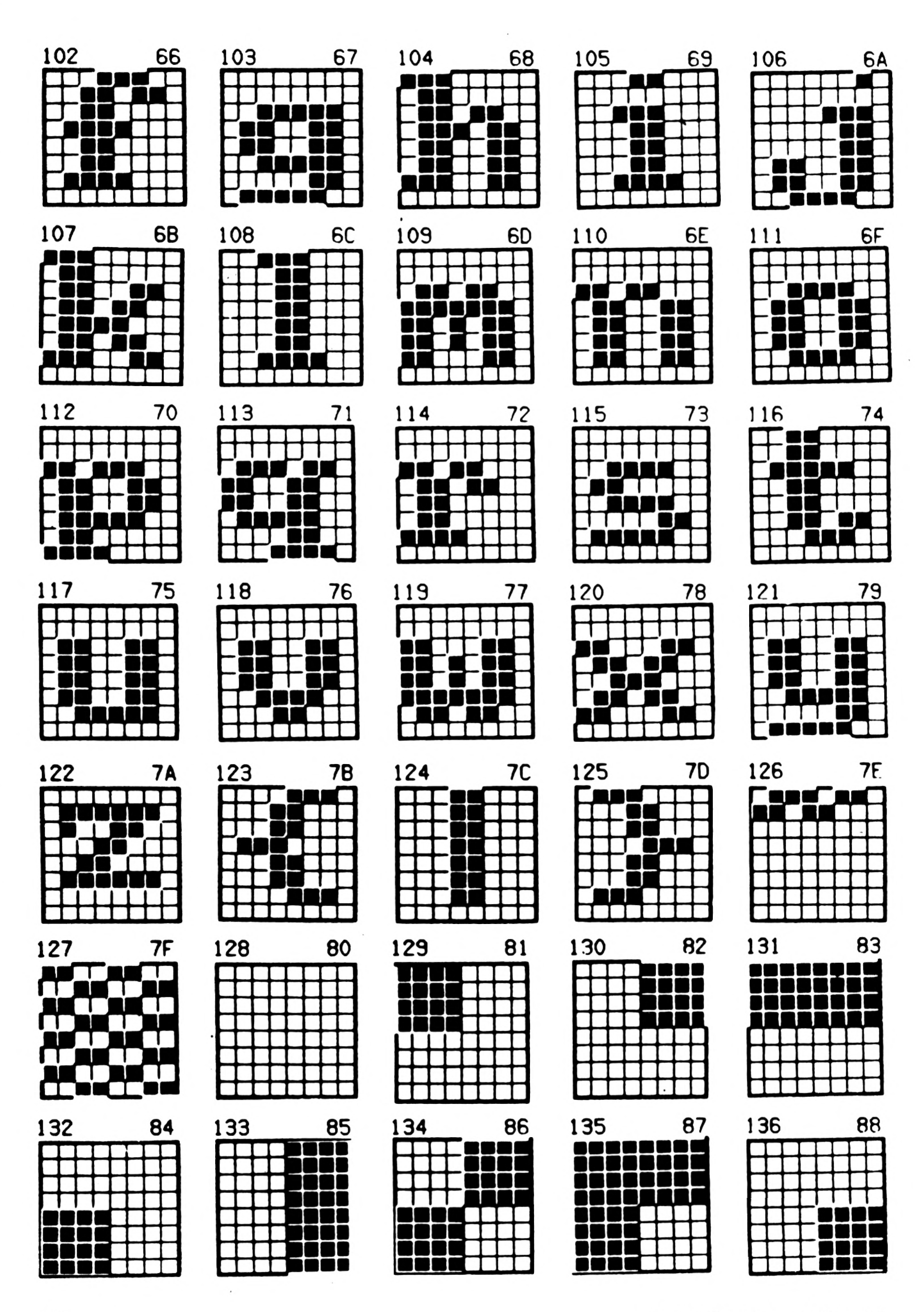
CÓDIGOS ASCII Y GRÁFICOS

El juego de caracteres del CPC464 representados en la matriz de 8 × 8 es el utilizado para la visualización en la pantalla. La orden **SYMBOL** permite redefinir los caracteres de nuevo.

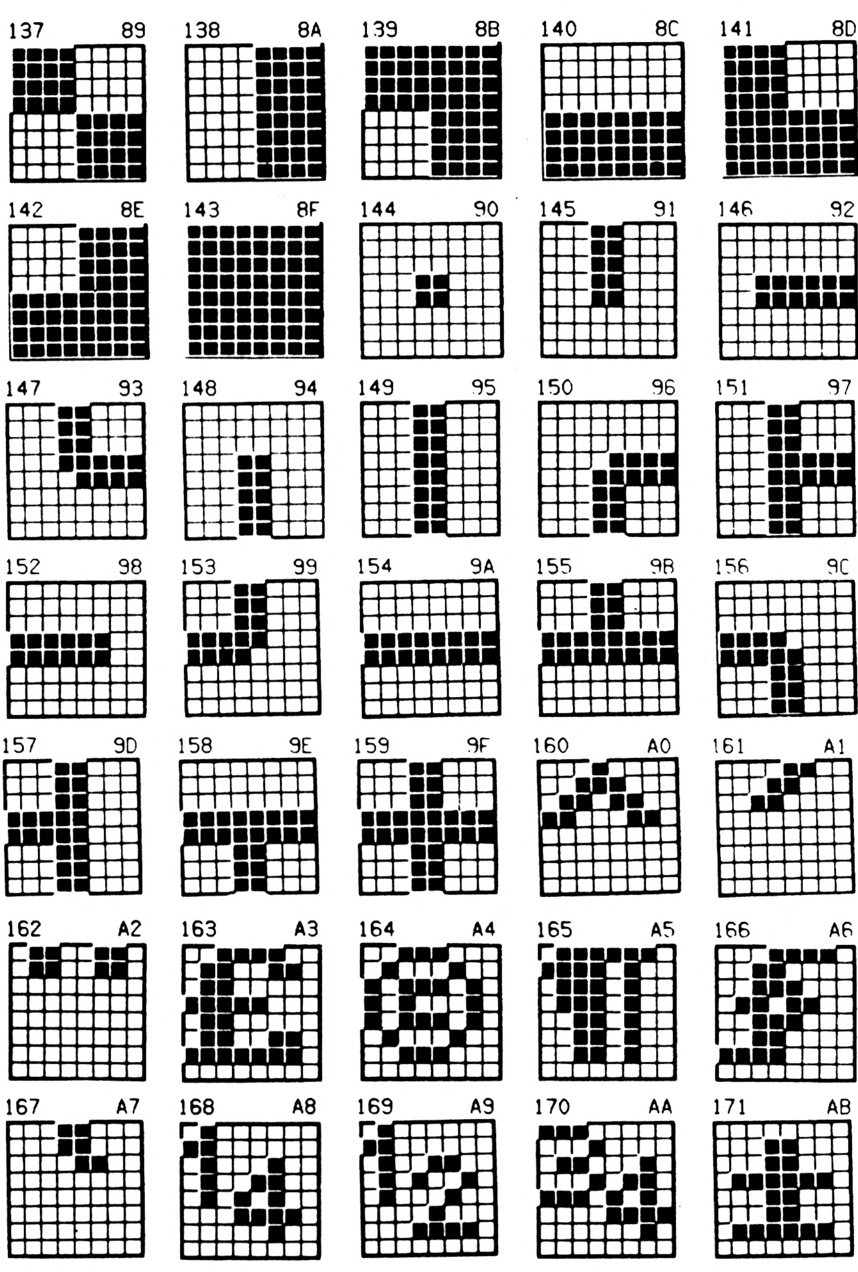


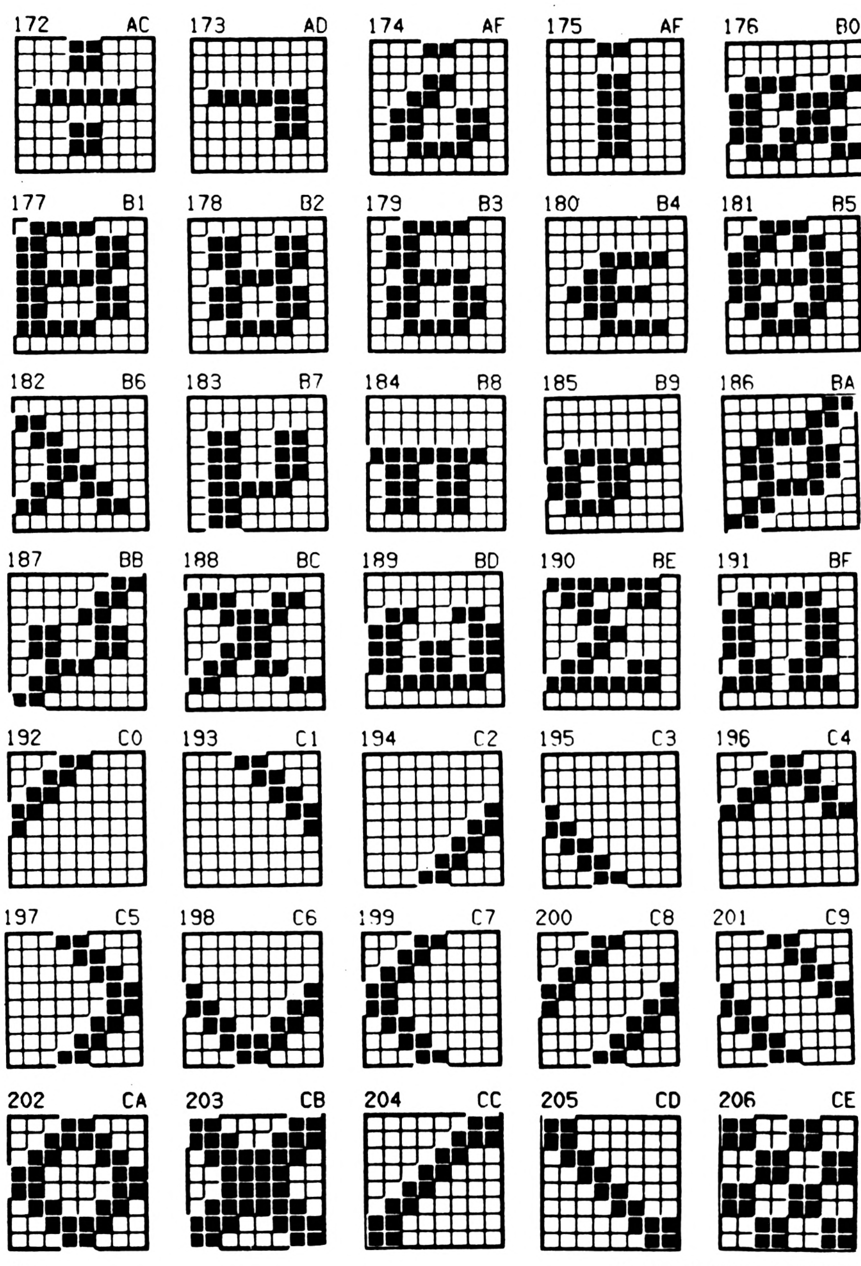

67 43
68 44
69 45
70 46
71 47
72 48
73 49
74 4A
75 4B
76 4C
77 4D
78 4E
79 4F
80 50
81 51
82 52
83 53
84 54
85 55
86 56
87 57
88 58
89 59
90 5A
91 5B
92 5C
93 5D
94 5E
95 5F
96 60
97 61
98 62
99 63
100 64
101 65

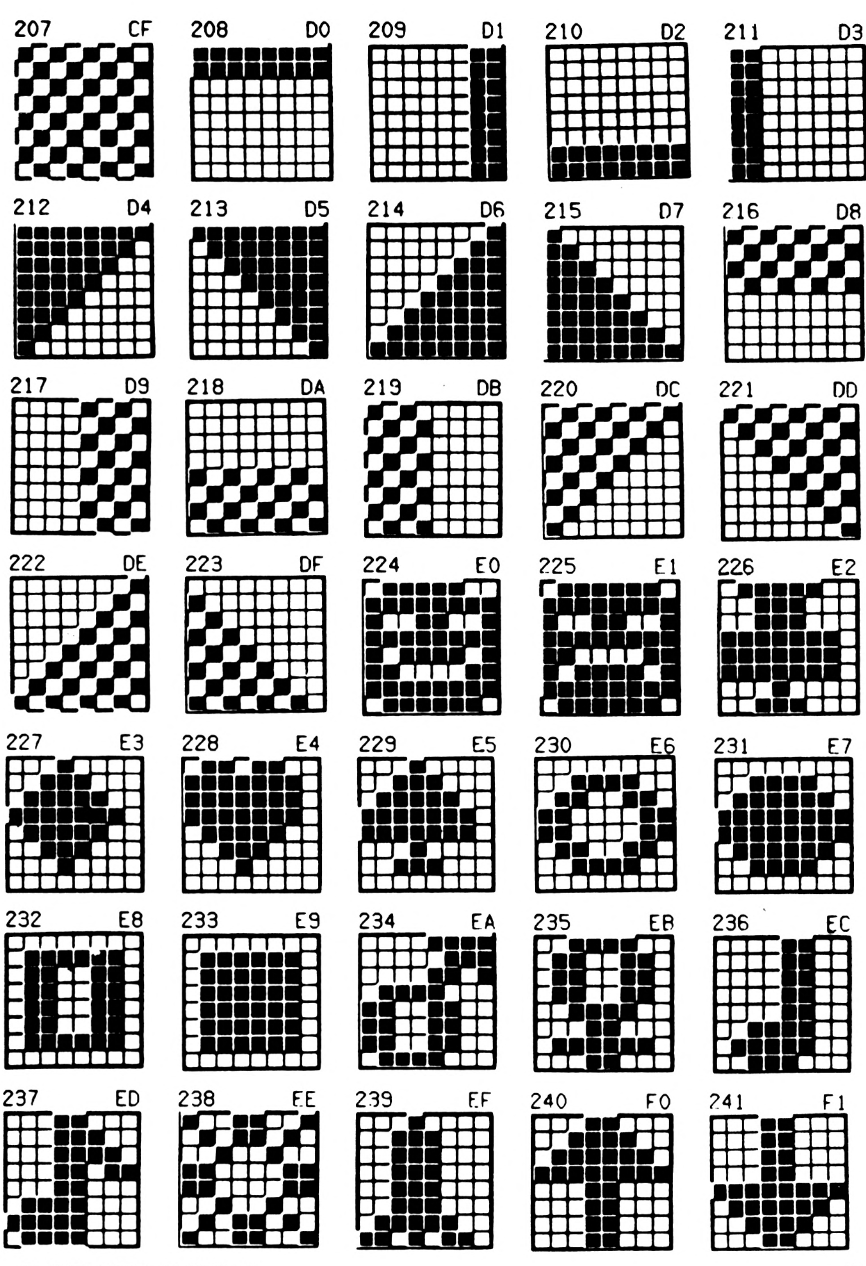
BASIC

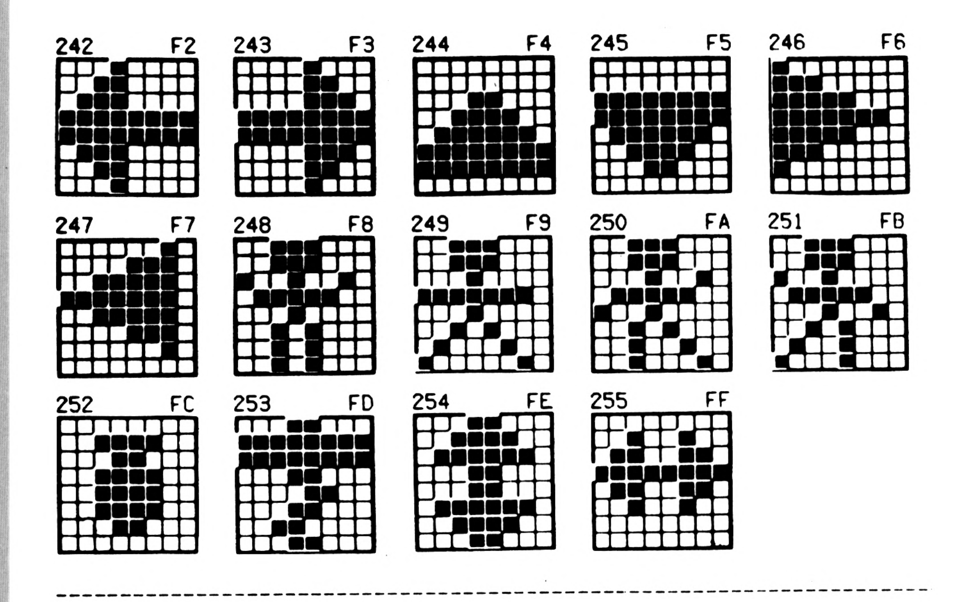


102 66
103 67
104 68
105 69
106 6A
107 6B
108 6C
109 6D
110 6E
111 6F
112 70
113 71
114 72
115 73
116 74
117 75
118 76
119 77
120 78
121 79
122 7A
123 7B
124 7C
125 7D
126 7E
127 7F
128 80
129 81
130 82
131 83
132 84
133 85
134 86
135 87
136 88


137 89
138 8A
139 8B
140 8C
141 8D
142 8E
143 8F
144 90
145 91
146 92
147 93
148 94
149 95
150 96
151 97
152 98
153 99
154 9A
155 9B
156 9C
157 9D
158 9E
159 9F
160 A0
161 A1
162 A2
163 A3
164 A4
165 A5
166 A6
167 A7
168 A8
169 A9
170 AA
171 AB


172 AC
173 AD
174 AF
175 AF
176 B0
177 B1
178 B2
179 B3
180 B4
181 B5
182 B6
183 B7
184 B8
185 B9
186 BA
187 BB
188 BC
189 BD
190 BE
191 BF
192 C0
193 C1
194 C2
195 C3
196 C4
197 C5
198 C6
199 C7
200 C8
201 C9
202 CA
203 CB
204 CC
205 CD
206 CE


207 CF
208 D0
209 D1
210 D2
211 D3
212 D4
213 D5
214 D6
215 D7
216 D8
217 D9
218 DA
219 DB
220 DC
221 DD
222 DE
223 DF
224 E0
225 E1
226 E2
227 E3
228 E4
229 E5
230 E6
231 E7
232 E8
233 E9
234 EA
235 EB
236 EC
237 ED
238 EE
239 EF
240 F0
241 F1


242 F2
243 F3
244 F4
245 F5
246 F6
247 F7
248 F8
249 F9
250 FA
251 FB
252 FC
253 FD
254 FE
255 FF

1 — **Unexpected NEXT**
NEXT no esperado. Se ha encontrado una orden NEXT sin que se haya ejecutado su correspondiente FOR.

2 — **Syntax error**
Error de sintaxis. El Basic no puede entender la línea.

3 — **Unexpected RETURN**
Se ha encontrado una orden RETURN sin que haya habido la orden GOSUB.

4 — **DATA exhausted**
Una orden READ ha intentado leer una línea de DATA que ya está agotada.

5 — **Improper argument**
El parámetro de una orden o el valor de una función no está expresado correctamente.

6 — **Overflow**
Desbordamiento. El valor introducido o calculado es demasiado grande o demasiado pequeño para que pueda ser manejado por el ordenador.

7 — **Memory full**
Memoria llena. Toda la memoria disponible ha sido utilizada o protegida. Esto puede suceder en caso de tablas muy grandes, de bucles FOR-NEXT o de llamadas GOSUB anidadas.

8 — **Line does not exist**
No existe en memoria el número de línea que se reclama.

9 — **Subscript out of range**
El índice de una tabla (array) está fuera de límites (demasiado grande o demasiado pequeño).

10 — **Array already dimensionned**
La tabla (array) ya había sido definida por DIM.

11 — **Division by zero**
Se ha intentado dividir un número por cero.

12 — **Invalid direct command**
No es válida en modo directo la orden tecleada.

13 — **Type mismatch**
Discordancia en el tipo. Tentativa de asignar un valor alfanumérico a una variable numérica o viceversa.

14 — **String space full**
Está lleno el espacio destinado a las variables de cadena.

15 — **String too long**
Una cadena alfanumérica comprende más de 255 caracteres.

16 — **String expression too complex**
Expresión alfanumérica demasiado compleja para ser manejada por el ordenador.

17 — **Cannot CONTinue**
No puede proseguir la ejecución del programa por medio de la orden CONT. Esto sucede si se ha modificado el programa después de una orden de parada (BREAK ESC).

18 — **Unknown user fonction**
Se ha olvidado definir la función FN con la orden DEF FN.

19 — **RESUME missing**
No se ha encontrado la orden RESUME en un subprograma llamado por la orden ON ERROR GOTO.

20 — **Unexpected RESUME**
Se ha encontrado una orden RESUME fuera de un subprograma llamado por la orden ON ERROR GOTO.

21 — **Direct command found**
Se ha encontrado una línea sin número, al cargar un programa en el cassette.

22 — **Operand missing**
No está el signo de la operación en la expresión que se ha hallado.

23 — **Line too long**
Línea demasiado larga. El Basic no acepta líneas de longitud superior a los 255 caracteres.

24 — **EOF met**
El programa ha llegado al "fin de fichero" de cassette.

25 — **File type error**
El fichero de cassette no es del tipo requerido.

26 — **NEXT missing**
No se ha encontrado el NEXT correspondiente a la orden FOR.

27 — **File already open**
Se ha intentado abrir un fichero que ya estaba abierto.

28 — **Unknown command**
Orden externa desconocida.

29 — **WEND missing**
El WEND correspondiente a la orden WHILE no se encuentra en el programa.

30 — **Unexpected WEND**
Se ha encontrado un WEND sin el WHILE precedente.

31 y siguientes — **Unknown error**
Error de tipo desconocido.

FORMATO DE ALMACENAMIENTO DE UNA LÍNEA BASIC EN MEMORIA

El ordenador sólo entiende códigos binarios. El intérprete Basic se encarga de traducir los programas al lenguaje binario con tal de que estén correctamente escritos.

Almacenamiento de las instrucciones Basic

A cada instrucción Basic, el intérprete le asigna un código llamado TOKEN. Este sistema ahorra un espacio considerable en la memoria central, puesto que es evidente que un byte ocupa menos espacio que una palabra entera. Ésta es la razón por la que no hay que usar nunca el nombre de las instrucciones Basic para definir las variables, ya que el intérprete las sustituiría por el TOKEN de la instrucción correspondiente.

Almacenamiento de una línea de Basic

El Basic comienza a almacenar las líneas a partir de la dirección 368. Veamos, por medio de un ejemplo, cómo almacena una línea.

Codificar el programilla siguiente:

```
1990 PRINT "HOLA"
```

Seguidamente, teclear directamente la secuencia de instrucciones siguientes:

```
FOR I=368 TO 390:PRINT I;"  ";PEEK(I):NEXT I
```

He aquí el conjunto de números que aparecerán, así como su significado.

Dirección	*Valor*	*Significado*
368	13	Byte inferior de la longitud de línea.
369	0	Byte superior de la longitud de línea.
370	198	Byte inferior del número de línea.
371	7	Byte superior del número de línea.
372	191	Código de la instrucción PRINT.
373	32	Código ASCII del espacio.
374	34	Código ASCII de ".
375	72	Código ASCII de H.
376	79	Código ASCII de O.
377	76	Código ASCII de L.
378	65	Código ASCII de A.
379	34	Código ASCII de ".
380	0	Código de separación.

FORMATO DE ALMACENAMIENTO DE UNA LÍNEA BASIC EN MEMORIA

La longitud de la línea viene expresada en dos bytes, y se obtiene por medio de la fórmula:

BYTE INFERIOR + (256 × BYTE SUPERIOR)

Por tanto, la longitud de la línea vale: 13 + (0 × 256) = 13. Efectivamente, constatamos que la línea ocupa 13 lugares de la memoria.

El número de línea está también expresado en dos bytes. Se obtiene por medio de la misma fórmula que para la longitud. Así, pues, vale:

198 + (256 × 7) = 1990

Para sustituir PRINT por REM, basta con codificar directamente: POKE 372,197. Si se lista se obtiene:

```
1990 REM "HOLA"
```

Desde ahora usted podrá modificar los programas a su gusto, o mejor, hacer que se modifiquen ellos mismos introduciendo líneas de POKE dentro del propio programa... ¡Que se divierta!

Veamos ahora cómo se almacenan las variables.
Codifique el programilla siguiente:

```
10 ABC=20
```

Seguidamente escriba como antes:

```
FOR I=368 TO 400:PRINT I; " ";PEEK(I):NEXT I
```

Verá que aparece:

Dirección	*Valor*	*Significado*
368	14	Byte inferior de la longitud de línea.
369	0	Byte superior de la longitud de línea.
370	10	Byte inferior del número de línea.
371	0	Byte superior del número de línea.
372	13	Indica una variable numérica.
373	7	Longitud del nombre de la variable +4.
374	0	Separación.
375	65	Código ASCII del primer carácter del nombre de la variable.
376	66	Código ASCII del segundo carácter del nombre de la variable.
377	195	128 + el código ASCII del último carácter del nombre de la variable.
378	239	TOKEN del signo =.
379	25	Magnitud de la variable.
380	20	Valor de la variable.
381	0	Separador.

El valor **13** en **372** es el código que indica que se trata de una variable numérica. Para una variable alfanumérica encontraríamos el valor 3.

De **375** a **377** encontramos codificado el nombre de la variable. Todos los caracteres están codificados en ASCII, excepto el último que está representado por su código ASCII aumentado en 128.

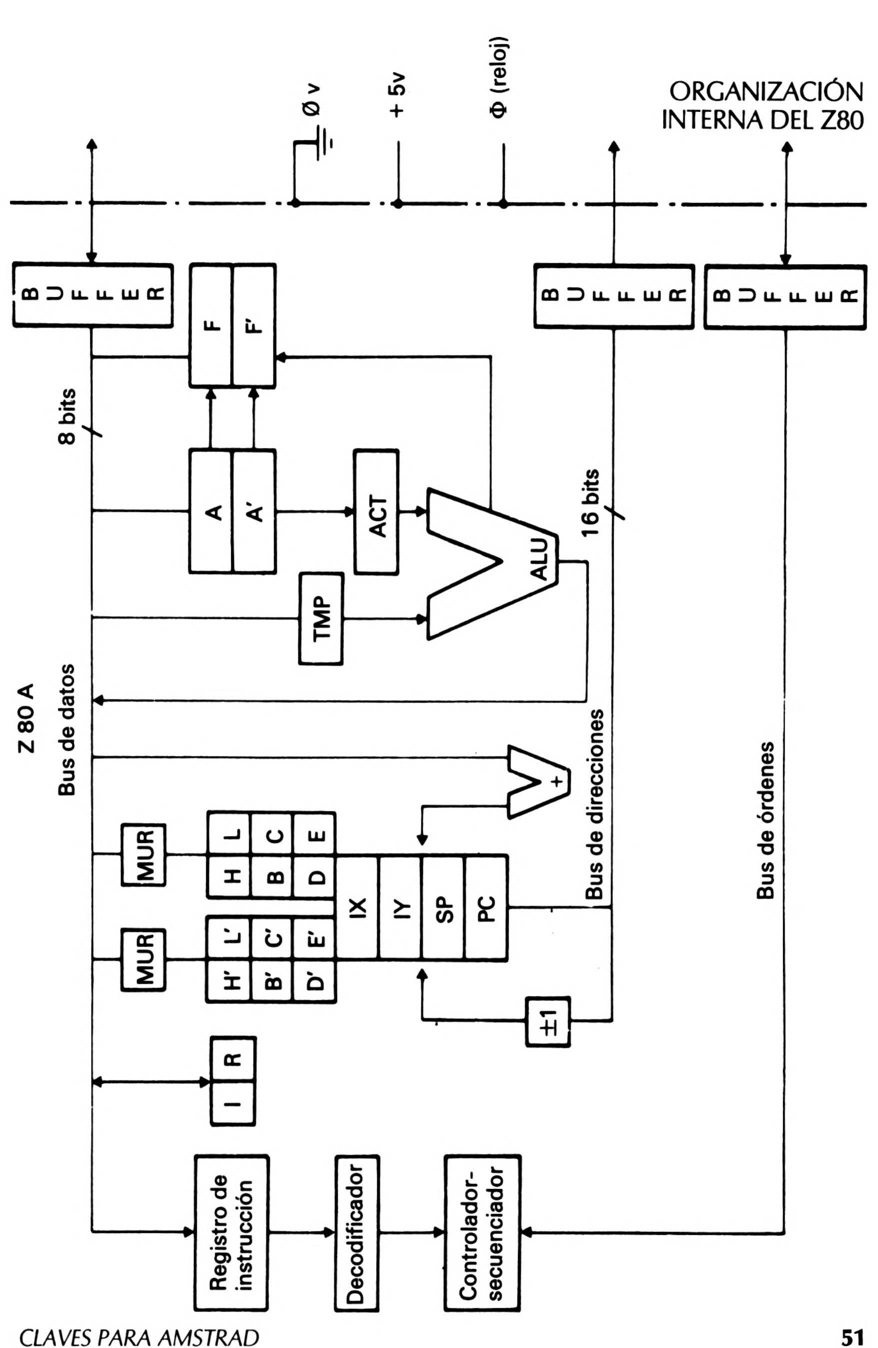
En **376**, el valor **239** representa el TOKEN del signo =. El TOKEN del signo = es distinto de su código ASCII. Así el ordenador sabe que el signo = no forma parte del nombre de la variable.

El valor **25** que se halla en **379** indica la magnitud de la variable. He aquí los distintos valores que se pueden encontrar en este lugar:

Valor	*Longitud de la variable*
15	Valor de la variable = 1, no está codificado.
16	Valor de la variable = 2, no está codificado.
..	.. .. ..
23	Valor de la variable = 9, no está codificado.
25	Valor de la variable comprendido entre 10 y 255, codificado en un byte.
26	Valor de la variable comprendido entre 255 y 65535, codificado en dos bytes.
31	Valor de la variable superior a 65535 o no entero, codificado en cinco bytes según la fórmula siguiente: VALOR=(2 (o5-145)*(65536+(o2/128)+(o3*2)+(o4*512)+(o1/32800) donde o1,...,o5 representan los valores presentes en las cinco direcciones que sirven para codificar la variable.

En el caso de que la variable sea un número negativo, el TOKEN del signo = (239) viene seguido del TOKEN del signo – (245).

ORGANIZACIÓN INTERNA DEL Z80



REGISTROS DEL Z80

Registros primarios

A (Acumulador)	F (Flag)
7 0	7 0
H	L
7 0	7 0
B	C
7 0	7 0
D	E
7 0	7 0

IX (Registro del índice)
15 0
IY (Registro del índice)
15 0
SP (Puntero de la pila)
15 0

I (Interrupciones)	R (Refresco)
7 0	7 0

Registros secundarios

A'	F'
H'	L'
B'	C'
D'	E'

PC (Contador ordinal)

Detalle del registro F (flag)

7	6	5	4	3	2	1	0
S	Z	-	H	-	P/V	N	C

S = Signo : se pone a 1 si el bit más significativo del resultado de una operación es 1.

Z = Cero : se pone a 1 si el resultado de la operación es nulo.

H = Mitad : idéntico a C, pero para las operaciones con medio byte o nibbles.

P/V = Paridad / desbord. : P=1 si hay un número par de bits puestos a 1, o V=1 si hay desbordamiento de capacidad tras una operación con números con signo.

N = Operación precedente : N=1 si la operación precedente fue una resta y N=0 si la operación precedente fue una suma.

C = Acarreo : se pone a 1 si el resultado produce un acarreo tanto si se trata de suma como de resta.

Nota: no pueden ser verificados los flags H y N.

Mnemónico	*Operación efectuada*
ADC	Suma con acarreo.
ADD	Suma sin acarreo.
BIT	Verifica un bit particular dentro de un byte.
CALL cc,nn	Llamada condicional a una subrutina.
CALL	Llamada incondicional a una subrutina.
CCF	Complementa el indicador de acarreo.
CP	Compara el operando y el acumulador.
CPD	Compara el contenido de la dirección apuntada por HL. Decrementa HL y BC.
CPDR	Compara el contenido de la dirección apuntada por HL. Decrementa HL y BC. Repite la secuencia hasta que BC=0.
CPI	Compara el contenido de la dirección apuntada por HL. Incrementa HL y decrementa BC.
CPIR	Compara el contenido de la dirección apuntada por HL. Incrementa HL y decrementa BC. Repite la secuencia hasta que BC=0.
CPL	Complementa el acumulador.
DAA	Ajuste decimal del acumulador.
DEC	Decrementa un registro, un par de registros, o una dirección apuntada por HL.
DI	Desactiva las interrupciones.
DJNZ	Decrementa B y efectúa un salto relativo si B≠0.
EI	Activa las interrupciones.
EX	Intercambia los contenidos de los registros.
EXX	Intercambia el contenido de los registros BC, DE y HL con los registros BC', DE' y HL'.
HALT	Coloca el microprocesador en posición de espera de una interrupción o de un reset.
IM	Posiciona uno de los tres modos de interrupción (de 0 a 2).
IN	Carga el acumulador o un registro con el contenido de una puerta de entrada/salida.
INC	Incrementa un registro, un par de registros, o el contenido de la dirección apuntada por HL.
IND	Carga la dirección apuntada por HL con el contenido de la puerta de entrada/salida apuntada por C, y decrementa HL y B.
INDR	Carga la dirección apuntada por HL con el contenido de la puerta de entrada/salida apuntada por C, y decrementa HL y B. Repite la secuencia hasta que B=0.

Mnemónico	*Operación efectuada*
INI	Carga la dirección apuntada por HL con el contenido de la puerta de entrada/salida apuntada por C, e incrementa HL y decrementa B.
INIR	Carga la dirección apuntada por HL con el contenido de la puerta de entrada/salida apuntada por C, e incrementa HL y decrementa B. Repite la secuencia hasta que B=0.
JP	Salto incondicional a la dirección dada o a la apuntada por HL, IX e IY.
JP cc,aa	Salto condicional (cc) a la dirección dada (aa).
JR e	Salto condicional (cc) relativo a PC más el desplazamiento (e).
JR cc,e	Salto condicional (cc) relativo a PC más el desplazamiento (e).
LD	Carga el acumulador, un registro o una dirección con el contenido del acumulador, un registro o una dirección.
LDD	Carga la dirección apuntada por HL con el contenido de la dirección apuntada por DE, después decrementa DE, HL y BC.
LDDR	Carga la dirección apuntada por HL con el contenido de la dirección apuntada por DE, después decrementa DE, HL y BC. Repite la secuencia hasta que BC=0.
LDI	Carga la dirección apuntada por HL con el contenido de la dirección apuntada por DE, después incrementa DE y HL, y decrementa BC.
LDIR	Carga la dirección apuntada por HL con el contenido de la dirección apuntada por DE, después incrementa DE y HL, y decrementa BC. Repite la secuencia hasta que BC=0.
NEG	Invierte el signo del acumulador.
NOP	El Z80 no efectúa instrucción alguna.
OR	"O" lógico entre el operando y el acumulador.
OTDR	Carga la puerta de entrada/salida apuntada por C con el contenido de la dirección apuntada por HL, después decrementa HL y B. Repite la secuencia hasta que B=0.
OTIR	Carga la puerta de entrada/salida apuntada por C con el contenido de la dirección apuntada por HL, después incrementa HL y decrementa B. Repite la secuencia hasta que B=0.
OUT	Carga la puerta de entrada/salida especificada con el contenido del acumulador.
OUTD	Carga la puerta de entrada/salida apuntada por C con el contenido de la dirección apuntada por HL, después decrementa HL y B.
OUTI	Carga la puerta de entrada/salida apuntada por C con el contenido de la dirección apuntada por HL, después incrementa HL y decrementa B.
POP	Carga un par de registros o un índice con el último valor de la pila (apuntada por SP).

Mnemónico	*Operación efectuada*
PUSH	Coloca el contenido de un par de registros o de un índice en la pila (apuntada por SP).
RES	Pone a 0 el bit especificado del operando.
RET	Retorno de una subrutina.
RETI	Retorno de una subrutina de interrupción.
RETN	Retorno de una subrutina de interrupción no enmascarable.
RL	Rotación a la izquierda mediante el acarreo del operando. C ← 7 ← 6 ← 5 ← 4 ← 3 ← 2 ← 1 ← 0 **Acarreo** **Operando 8 bits**
RLA	Rotación a la izquierda mediante el acarreo del acumulador. C ← 7 ← 6 ← 5 ← 4 ← 3 ← 2 ← 1 ← 0 **Acarreo** **Acumulador**
RLC	Rotación circular a la izquierda del contenido de un registro o de una dirección apuntada por HL o por IX e IY más desplazamiento. C ← 7 ← 6 ← 5 ← 4 ← 3 ← 2 ← 1 ← 0 **Acarreo** **Operando 8 bits**
RLCA	Rotación circular del acumulador a la izquierda. C ← 7 ← 6 ← 5 ← 4 ← 3 ← 2 ← 1 ← 0 **Acarreo** **Acumulador**
RLD	Rotación BCD (binario codificado decimal) de 4 bits a la izquierda entre el acumulador y la dirección apuntada por HL. 7 6 5 4 | 3 2 1 0 7 6 5 4 | 3 2 1 0 **Acumulador** **Dirección apuntada por HL**

Mnemónico	*Operación efectuada*
RR	Rotación a la derecha mediante el acarreo del operando. 7 → 6 → 5 → 4 → 3 → 2 → 1 → 0 → C **Operando 8 bits** **Acarreo**
RRA	Rotación a la derecha mediante el acarreo del acumulador. 7 → 6 → 5 → 4 → 3 → 2 → 1 → 0 → C **Acumulador** **Acarreo**
RRC	Rotación circular del operando a la derecha. 7 → 6 → 5 → 4 → 3 → 2 → 1 → 0 → C **Operando 8 bits** **Acarreo**
RRCA	Rotación circular del acumulador a la derecha. 7 → 6 → 5 → 4 → 3 → 2 → 1 → 0 → C **Acumulador** **Acarreo**
RRD	Rotación BCD (binario codificado decimal) a la derecha de 4 bits entre el acumulador y el contenido de la dirección apuntada por HL. 7 6 5 4 3 2 1 0 — 7 6 5 4 3 2 1 0 **Acumulador** **Dirección apuntada por HL**
RST	Salto a una dirección de un byte.
SBC	Resta con acarreo entre el acumulador y el operando o HL y un par de registros.
SCF	Pone a 1 el flag de acarreo.
SET	Pone a 1 un bit particular de un registro o de una dirección apuntada por HL o IX e IY más desplazamiento.
SLA	Desplazamiento aritmético a la izquierda del operando. C ← 7 ← 6 ← 5 ← 4 ← 3 ← 2 ← 1 ← 0 ← Ø **Acarreo** **Operando 8 bits**

Mnemónico	*Operación efectuada*
SRA	Desplazamiento aritmético a la derecha del operando. 7 → 6 → 5 → 4 → 3 → 2 → 1 → 0 --► C **Operando 8 bits** **Acarreo** *Nota:* el bit 7 permanece sin cambiar.
SRL	Desplazamiento lógico a la derecha del operando. Ø --► 7 → 6 → 5 → 4 → 3 → 2 → 1 → 0 --► C **Operando 8 bits** **Acarreo**
SUB	Resta el operando del acumulador.
XOR	"O" exclusivo entre el operando y el acumulador.

- d = dato en 8 bits;
- dd = dato en 16 bits;
- aa = dirección en 16 bits.

- ● = flag modificado
- 0 = flag a 0
- 1 = flag a 1

Código objeto	*Instrucción*	*S*	*Z*	*P/V*	*C*
8E	ADC A,(HL)	●	●	●	●
DD8Ed	ADC A,(IX+d)	●	●	●	●
FD8Ed	ADC A,(IY+d)	●	●	●	●
8F	ADC A,A	●	●	●	●
88	ADC A,B	●	●	●	●
89	ADC A,C	●	●	●	●
8A	ADC A,D	●	●	●	●
8B	ADC A,E	●	●	●	●
8C	ADC A,H	●	●	●	●
8D	ADC A,L	●	●	●	●
CEd	ADC A,d	●	●	●	●
ED4A	ADC HL,BC	●	●	●	●
ED5A	ADC HL,DE	●	●	●	●
ED6A	ADC HL,HL	●	●	●	●
ED7A	ADC HL,SP	●	●	●	●
86	ADD A,(HL)	●	●	●	●
DD86d	ADD A,(IX+d)	●	●	●	●
FD86d	ADD A,(IY+d)	●	●	●	●
87	ADD A,A	●	●	●	●
80	ADD A,B	●	●	●	●
81	ADD A,C	●	●	●	●
82	ADD A,D	●	●	●	●
83	ADD A,E	●	●	●	●
84	ADD A,H	●	●	●	●
85	ADD A,L	●	●	●	●
C6d	ADD A,d	●	●	●	●
09	ADD HL,BC				●
19	ADD HL,DE				●
29	ADD HL,HL				●
39	ADD HL,SP				●
DD09	ADD IX,BC				●
DD19	ADD IX,DE				●
DD29	ADD IX,IX				●
DD39	ADD IX,SP				●
FD09	ADD IY,BC				●
FD19	ADD IY,DE				●
FD29	ADD IY,IY				●
FD39	ADD IY,SP				●
A6	AND (HL)	●	●	●	0
DDA6d	AND (IX+d)	●	●	●	0
FDA6d	AND (IY+d)	●	●	●	0
A7	AND A	●	●	●	0

CÓDIGOS DE LAS INSTRUCCIONES DEL Z80 POR ORDEN ALFABÉTICO

Código objeto	*Instrucción*		*S*	*Z*	*P/V*	*C*
A0	AND	B	●	●	●	0
A1	AND	C	●	●	●	0
A2	AND	D	●	●	●	0
A3	AND	E	●	●	●	0
A4	AND	H	●	●	●	0
A5	AND	L	●	●	●	0
E6d	AND	d	●	●	●	0
CB46	BIT	0,(HL)	●	●	●	
DDCBd46	BIT	0,(IX+d)	●	●	●	
FDCBd46	BIT	0,(IY+d)	●	●	●	
CB47	BIT	0,A	●	●	●	
CB40	BIT	0,B	●	●	●	
CB41	BIT	0,C	●	●	●	
CB42	BIT	0,D	●	●	●	
CB43	BIT	0,E	●	●	●	
CB44	BIT	0,H	●	●	●	
CB45	BIT	0,L	●	●	●	
CB4E	BIT	1 (HL)	●	●	●	
DDCBd4E	BIT	1,(IX+d)	●	●	●	
FDCBd4E	BIT	1,(IY+d)	●	●	●	
CB4F	BIT	1,A	●	●	●	
CB48	BIT	1,B	●	●	●	
CB49	BIT	1,C	●	●	●	
CB4A	BIT	1,D	●	●	●	
CB4B	BIT	1,E	●	●	●	
CB4C	BIT	1,H	●	●	●	
CB4D	BIT	1,L	●	●	●	
CB56	BIT	2,(HL)	●	●	●	
DDCBd56	BIT	2,(IX+d)	●	●	●	
FDCBd56	BIT	2,(IY+d)	●	●	●	
CB57	BIT	2,A	●	●	●	
CB50	BIT	2,B	●	●	●	
CB51	BIT	2,C	●	●	●	
CB52	BIT	2,D	●	●	●	
CB53	BIT	2,E	●	●	●	
CB54	BIT	2,H	●	●	●	
CB55	BIT	2,L	●	●	●	
CB5E	BIT	3,(HL)	●	●	●	
DDCBd5E	BIT	3,(IX+d)	●	●	●	
FDCBd5E	BIT	3,(IY+d)	●	●	●	
CB5F	BIT	3,A	●	●	●	
CB58	BIT	3,B	●	●	●	
CB59	BIT	3,C	●	●	●	
CB5A	BIT	3,D	●	●	●	
CB5B	BIT	3,E	●	●	●	
CB5C	BIT	3,H	●	●	●	
CB5D	BIT	3,L	●	●	●	

Código objeto	Instrucción	S	Z	P/V	C
CB66	BIT 4,(HL)	●	●	●	
DDCBd66	BIT 4,(IX+d)	●	●	●	
FDCBd66	BIT 4,(IY+d)	●	●	●	
CB67	BIT 4,A	●	●	●	
CB60	BIT 4,B	●	●	●	
CB61	BIT 4,C	●	●	●	
CB62	BIT 4,D	●	●	●	
CB63	BIT 4,E	●	●	●	
CB64	BIT 4,H	●	●	●	
CB65	BIT 4,L	●	●	●	
CB6E	BIT 5,(HL)	●	●	●	
DDCBd6E	BIT 5,(IX+d)	●	●	●	
FDCBd6E	BIT 5,(IY+d)	●	●	●	
CB6F	BIT 5,A	●	●	●	
CB68	BIT 5,B	●	●	●	
CB69	BIT 5,C	●	●	●	
C86A	BIT 5,D	●	●	●	
CB6B	BIT 5,E	●	●	●	
CB6C	BIT 5,H	●	●	●	
CB6D	BIT 5,L	●	●	●	
DB76	BIT 6,(HL)	●	●	●	
DDCBd76	BIT 6,(IX+d)	●	●	●	
FDCBd76	BIT 6,(IY+d)	●	●	●	
CB77	BIT 6,A	●	●	●	
CB70	BIT 6,B	●	●	●	
CB71	BIT 6,C	●	●	●	
CB72	BIT 6,D	●	●	●	
CB73	BIT 6,E	●	●	●	
CB74	BIT 6,H	●	●	●	
CB75	BIT 6,L	●	●	●	
CB7E	BIT 7,(HL)	●	●	●	
DDCBd7E	BIT 7,(IX+d)	●	●	●	
FDCBd7E	BIT 7,(IY+d)	●	●	●	
CB7F	BIT 7,A	●	●	●	
CB78	BIT 7,B	●	●	●	
CB79	BIT 7,C	●	●	●	
CB7A	BIT 7,D	●	●	●	
CB7B	BIT 7,E	●	●	●	
CB7C	BIT 7,H	●	●	●	
CB7D	BIT 7,L	●	●	●	
DCaa	CALL C,aa				
FCaa	CALL M,aa				
D4aa	CALL NC,aa				
C4aa	CALL NZ,aa				
F4aa	CALL P,aa				
ECaa	CALL PE,aa				

CÓDIGOS DE LAS INSTRUCCIONES DEL Z80
POR ORDEN ALFABÉTICO

Código objeto	*Instrucción*	*S*	*Z*	*P/V*	*C*
E4aa	CALL PO,aa				
CCaa	CALL Z;aa				
CDaa	CALL aa				
3F	CCF				●
BE	CP (HL)	●	●	●	●
DDBEd	CP (IX+d)	●	●	●	●
FDBEd	CP (IY+d)	●	●	●	●
BF	CP A	●	●	●	●
B8	CP B	●	●	●	●
B9	CP C	●	●	●	●
BA	CP D	●	●	●	●
BB	CP E	●	●	●	●
BC	CP H	●	●	●	●
BD	CP L	●	●	●	●
FEd	CP d	●	●	●	●
EDA9	CPD	●	●	●	
EDB9	CPDR	●	●	●	
EDB1	CPIR	●	●	●	
EDA1	CPI	●	●	●	
2F	CPL				
27	DAA	●	●	●	
35	DEC (HL)	●	●	●	
DD35d	DEC (IX+d)	●	●	●	
FD35d	DEC (IY+d)	●	●	●	
3D	DEC A	●	●	●	
05	DEC B	●	●	●	
0B	DEC BC				
0D	DEC C	●	●	●	
15	DEC D	●	●	●	
1B	DEC DE				
1D	DEC E	●	●	●	
25	DEC H	●	●	●	
2B	DEC HL				
DD2B	DEC IX				
FD2B	DEC IY				
2D	DEC L	●	●	●	
3B	DEC SP				
F3	DI				
10d	DJNZ d				
FB	EI				
E3	EX (SP),HL				
DDE3	EX (SP),IX				
FDE3	EX (SP),IY				
08	EX AF,AF'				
EB	EX DE,HL				
D9	EXX				

Código objeto	*Instrucción*	*S*	*Z*	*P/V*	*C*
76	HALT				
ED46	IM 0				
ED56	IM 1				
ED5E	IM 2				
ED78	IN A,(C)	●	●	●	
ED40	IN B,(C)	●	●	●	
ED48	IN C,(C)	●	●	●	
ED50	IN D,(C)	●	●	●	
ED58	IN E,(C)	●	●	●	
ED60	IN H,(C)	●	●	●	
ED68	IN L,(C)	●	●	●	
DBd	IN A,(d)				
34	INC (HL)	●	●	●	
DD34d	INC (IX+d)	●	●	●	
FD34d	INC (IY+d)	●	●	●	
3C	INC A	●	●	●	
04	INC B	●	●	●	
03	INC BC				
0C	INC C	●	●	●	
14	INC D	●	●	●	
13	INC DE				
1C	INC E	●	●	●	
24	INC H	●	●	●	
23	INC HL				
DD23	INC IX				
FD23	INC IY				
2C	INC L	●	●	●	
33	INC SP				
EDAA	IND	●	●	●	
EDBA	INDR	●	●	●	
EDA2	INI	●	●	●	
EDB2	INIR	●	●	●	
C3aa	JP aa				
E9	JP (HL)				
DDE9	JP (IX)				
FDE9	JP (IY)				
DAdd	JP C,aa				
FAaa	JP M,aa				
D2aa	JP NC,aa				
C2aa	JP NZ.aa				
F2aa	JP P,aa				
EAaa	JP PE,aa				
E2aa	JP PO,aa				
CAaa	JP Z,aa				
38d	JR C,d				
30d	JR NC,d				
20d	JR NZ,d				

CÓDIGOS DE LAS INSTRUCCIONES DEL Z80 POR ORDEN ALFABÉTICO

Código objeto	*Instrucción*		*S*	*Z*	*P/V*	*C*
28d	JR	Z,d				
18d	JR	d				
02	LD	(BC),A				
12	LD	(DE),A				
77	LD	(HL),A				
70	LD	(HL),B				
71	LD	(HL),C				
72	LD	(HL),D				
73	LD	(HL),E				
74	LD	(HL),H				
75	LD	(HL),L				
36d	LD	(HL),d				
DD77d	LD	(IX+d),A				
DD70d	LD	(IX+d),B				
DD71d	LD	(IX+d),C				
DD72d	LD	(IX+d),D				
DD73d	LD	(IX+d),E				
DD74d	LD	(IX+d),H				
DD75d	LD	(IX+d),L				
DD36d20	LD	(IX+d),d				
FD77d	LD	(IY+d),A				
FD70d	LD	(IY+d),B				
FD71d	LD	(IY+d),C				
FD72d	LD	(IY+d),D				
FD73d	LD	(IY+d),E				
FD74d	LD	(IY+d),H				
FD75d	LD	(IY+d),L				
FD36d20	LD	(IY+d),d				
32dd	LD	(dd),A				
ED43dd	LD	(dd),BC				
ED53dd	LD	(dd),DE				
22dd	LD	(dd),HL				
DD22dd	LD	(dd),IX				
FD22dd	LD	(dd),IY				
ED73dd	LD	(dd),SP				
0A	LD	A,(BC)				
1A	LD	A,(DE)				
7E	LD	A,(HL)				
DD7Ed	LD	A,(IX+d)				
FD7Ed	LD	A,(IY+d)				
3Add	LD	A,(dd)				
7F	LD	A,A				
78	LD	A,B				
79	LD	A,C				
7A	LD	A,D				
7B	LD	A,E				
7C	LD	A,H				

Código objeto	*Instrucción*	*S*	*Z*	*P/V*	*C*
ED57	LD A,I	●	●	●	
7D	LD A,L				
3E	LD A,d				
ED5F	LD A,R	●	●	●	
46	LD B,(HL)				
DD46d	LD B,(IX+d)				
FD46d	LD B,(IY+d)				
47	LD B,A				
40	LD B,B				
41	LD B,C				
42	LD B,D				
43	LD B,E				
44	LD B,H				
45	LD B,L				
06d	LD B,d				
ED4Bdd	LD BC,(dd)				
01dd	LD BC,dd				
4E	LD C,(HL)				
DD4Ed	LD C,(IX+d)				
FD4Ed	LD C,(IY+d)				
4F	LD C,A				
48	LD C,B				
49	LD C,C				
4A	LD C,D				
4B	LD C,E				
4C	LD C,H				
4D	LD C,L				
0Ed	LD C,d				
56	LD D,(HL)				
DD56d	LD D,(IX+d)				
FD56d	LD D,(IY+d)				
57	LD D,A				
50	LD D,B				
51	LD D,C				
52	LD D,D				
53	LD D,E				
54	LD D,H				
55	LD D,L				
16d	LD D,d				
ED5Bdd	LD DE,(dd)				
11dd	LD DE,dd				
5E	LD E,(HL)				
DD5Ed	LD E,(IX+d)				
FD5Ed	LD E,(IY+d)				
5F	LD E,A				
58	LD E,B				
59	LD E,C				

CÓDIGOS DE LAS INSTRUCCIONES DEL Z80
POR ORDEN ALFABÉTICO

Código objeto	*Instrucción*		*S*	*Z*	*P/V*	*C*
5A	LD	E,D				
5B	LD	E,E				
5C	LD	E,H				
5D	LD	E,L				
1E20	LD	E,n				
66	LD	H,(HL)				
DD66d	LD	H,(IX+d)				
FD66d	LD	H,(IY+d)				
67	LD	H,A				
60	LD	H,B				
61	LD	H,C				
62	LD	H,D				
63	LD	H,E				
64	LD	H,H				
65	LD	H,L				
26d	LD	H,d				
2Add	LD	HL,(dd)				
21dd	LD	HL,dd				
ED47	LD	I,A				
DD2Add	LD	IX,(dd)				
DD21dd	LD	IX,dd				
FD2Add	LD	IY,(dd)				
FD21dd	LD	IY,dd				
6E	LD	L,(HL)				
DD6Ed	LD	L,(IX+d)				
FD6Ed	LD	L,(IY+d)				
6F	LD	L,A				
68	LD	L,B				
69	LD	L,C				
6A	LD	L,D				
6B	LD	L,E				
6C	LD	L,H				
6D	LD	L,L				
2Ed	LD	L,d				
ED4F	LD	R,A				
ED7Bdd	LD	SP,(dd)				
F9	LD	SP,HL				
DDF9	LD	SP,IX				
FDF9	LD	SP,IY				
31dd	LD	SP,dd				
EDA8	LDD				●	
EDB8	LDDR				0	
EDA0	LDI				●	
EDB0	LDIR				0	
ED44	NEG					
00	NOP					
B6	OR	(HL)	●	●	●	0

Código objeto	*Instrucción*		*S*	*Z*	*P/V*	*C*
DDB6d	OR	(IX+d)	●	●	●	0
FDB6d	OR	(IY+d)	●	●	●	0
B7	OR	A	●	●	●	0
B0	OR	B	●	●	●	0
B1	OR	C	●	●	●	0
B2	OR	D	●	●	●	0
B3	OR	E	●	●	●	0
B4	OR	H	●	●	●	0
B5	OR	L	●	●	●	0
F6d	OR	d	●	●	●	0
EDBB	OTDR		●	●	●	
EDB3	OTIR		●	●	●	
ED79	OUT	(C),A				
ED41	OUT	(C),B				
ED49	OUT	(C),C				
ED51	OUT	(C),D				
ED59	OUT	(C),E				
ED61	OUT	(C),H				
ED69	OUT	(C),L				
D3d	OUT	(d),A				
EDAB	OUTD		●	●	●	
EDA3	OUTI		●	●	●	
F1	POP	AF				
C1	POP	BC				
D1	POP	DE				
E1	POP	HL				
DDE1	POP	IX				
FDE1	POP	IY				
F5	PUSH	AF				
C5	PUSH	BC				
D5	PUSH	DE				
E5	PUSH	HL				
DDE5	PUSH	IX				
FDE5	PUSH	IY				
CB86	RES	0,(HL)				
DDCBd86	RES	0,(IX+d)				
FDCBd86	RES	0,(IY+d)				
CB87	RES	0,A				
CB80	RES	0,B				
CB81	RES	0,C				
CB82	RES	0,D				
CB83	RES	0,E				
CB84	RES	0,H				
CB85	RES	0,L				
CB8E	RES	1,(HL)				
DDCBd8E	RES	1,(IX+d)				

Código objeto	*Instrucción*	*S*	*Z*	*P/V*	*C*
FDCBd8E	RES 1,(IY+d)				
CB8F	RES 1,A				
CB88	RES 1,B				
CB89	RES 1,C				
CB8A	RES 1,D				
CB8B	RES 1,E				
CB8C	RES 1,H				
CB8D	RES 1,L				
CB96	RES 2,(HL)				
DDCBd96	RES 2,(IX+d)				
FDCBd96	RES 2,(IY+d)				
CB97	RES 2,A				
CB90	RES 2,B				
CB91	RES 2,C				
CB92	RES 2,D				
CB93	RES 2,E				
CB94	RES 2,H				
CB95	RES 2,L				
CB9E	RES 3,(HL)				
DDCBd9E	RES 3,(IX+d)				
FDCBd9E	RES 3,(IY+d)				
CB9F	RES 3,A				
CB98	RES 3,B				
CB99	RES 3,C				
CB9A	RES 3,D				
CB9B	RES 3,E				
CB9C	RES 3,H				
CB9D	RES 3,L				
CBA6	RES 4,(HL)				
DDCBdA6	RES 4,(IX+d)				
FDCBdA7	RES 4,(IY+d)				
CBA7	RES 4,A				
CBA0	RES 4,B				
CBA1	RES 4,C				
CBA2	RES 4,D				
DBA3	RES 4,E				
CBA4	RES 4,H				
CBA5	RES 4,L				
CBAE	RES 5,(HL)				
DDCBdAE	RES 5,(IX+d)				
FDCBdAE	RES 5,(IY+d)				
CBAF	RES 5,A				
CBA8	RES 5,B				
CBA9	RES 5,C				
CBAA	RES 5,D				
CBAB	RES 5,E				
CBAC	RES 5,H				

Código objeto	Instrucción		S	Z	P/V	C
CBAD	RES	5,L				
CBB6	RES	6,(HL)				
DDCBdB6	RES	6,(IX+d)				
FDCBdB6	RES	6,(IY+d)				
CBB7	RES	6,A				
CBB0	RES	6,B				
CBB1	RES	6,C				
CBB2	RES	6,D				
CBB3	RES	6,E				
CBB4	RES	6,H				
CBB5	RES	6,L				
CBBE	RES	7,(HL)				
DDCBdBE	RES	7,(IX+d)				
FDCBdBE	RES	7,(IY+d)				
CBBF	RES	7,A				
CBB8	RES	7,B				
CBB9	RES	7,C				
CBBA	RES	7,D				
CBBB	RES	7,E				
CBBC	RES	7,H				
CBBD	RES	7,L				
C9	RET					
D8	RET	C				
F8	RET	M				
D0	RET	NC				
C0	RET	NZ				
F0	RET	P				
E8	RET	PE				
E0	RET	PO				
C8	RET	Z				
ED4D	RETI					
ED45	RETN					
CB16	RL	(HL)	●	●	●	●
DDCBd16	RL	(IX+d)	●	●	●	●
FDCBd16	RL	(IY+d)	●	●	●	●
CB17	RL	A	●	●	●	●
CB10	RL	B	●	●	●	●
CB11	RL	C	●	●	●	●
CB12	RL	D	●	●	●	●
CB13	RL	E	●	●	●	●
CB14	RL	H	●	●	●	●
CB15	RL	L	●	●	●	●
17	RLA					●
CB06	RLC	(HL)	●	●	●	●
DDCBd06	RLC	(IX+d)	●	●	●	●
FDCBd06	RLC	(IY+d)	●	●	●	●

CÓDIGOS DE LAS INSTRUCCIONES DEL Z80 POR ORDEN ALFABÉTICO

Código objeto	*Instrucción*	*S*	*Z*	*P/V*	*C*
CB07	RLC A	●	●	●	●
CB00	RLC B	●	●	●	●
CB01	RLC C	●	●	●	●
CB02	RLC D	●	●	●	●
CB03	RLC E	●	●	●	●
CB04	RLC H	●	●	●	●
CB05	RLC L	●	●	●	●
07	RLCA				●
ED6F	RLD	●	●	●	
CB1E	RR (HL)	●	●	●	●
DDCBd1E	RR (IX+d)	●	●	●	●
FDCBd1E	RR (IY+d)	●	●	●	●
CB1F	RR A	●	●	●	●
CB18	RR B	●	●	●	●
CB19	RR C	●	●	●	●
CB1A	RR D	●	●	●	●
CB1B	RR E	●	●	●	●
CB1C	RR H	●	●	●	●
CB1D	RR L	●	●	●	●
1F	RRA				●
CB0E	RRC (HL)	●	●	●	●
DDCBd0E	RRC (IX+d)	●	●	●	●
FDCBd0E	RRC (IY+d)	●	●	●	●
CB0F	RRC A	●	●	●	●
CB08	RRC B	●	●	●	●
CB09	RRC C	●	●	●	●
CB0A	RRC D	●	●	●	●
CB0B	RRC E	●	●	●	●
CB0C	RRC H	●	●	●	●
CB0D	RRC L	●	●	●	●
0F	RRCA				●
ED67	RRD	●	●	●	
C7	RST 00H				
CF	RST 08H				
D7	RST 10H				
DF	RST 18H				
E7	RST 20H				
EF	RST 28H				
F7	RST 30H				
FF	RST 38H				
DEd	SBC A,d	●	●	●	●
9E	SBC A,(HL)	●	●	●	●
DD9Ed	SBC A,(IX+d)	●	●	●	●
FD9Ed	SBC A,(IY+d)	●	●	●	●
9F	SBC A,A	●	●	●	●
98	SBC A,B	●	●	●	●

Código objeto	*Instrucción*	*S*	*Z*	*P/V*	*C*
99	SBC A,C	●	●	●	●
9A	SBC A,D	●	●	●	●
9B	SBC A,E	●	●	●	●
9C	SBC A,H	●	●	●	●
9D	SBC A,L	●	●	●	●
ED42	SBC HL,BC	●	●	●	●
ED52	SBC HL,DE	●	●	●	●
ED62	SBC HL,HL	●	●	●	●
ED72	SBC HL,SP	●	●	●	●
37	SCF				1
CBC6	SET 0,(HL)				
DDCBdC6	SET 0,(IX+d)				
FDCBdC6	SET 0,(IY+d)				
CBC7	SET 0,A				
CBC0	SET 0,B				
CBC1	SET 0,C				
CBC2	SET 0,D				
CBC3	SET 0,E				
CBC4	SET 0,H				
CBC5	SET 0,L				
CBCE	SET 1,(HL)				
ddCBdCE	SET 1,(IX+d)				
FDCBdCE	SET 1,(IY+d)				
CBCF	SET 1,A				
CBC8	SET 1,B				
CBC9	SET 1,C				
CBCA	SET 1,D				
CBCB	SET 1,E				
CBCC	SET 1,H				
CBCD	SET 1,L				
CBD6	SET 2,(HL)				
DDCBdD6	SET 2,(IX+d)				
FDCBdD6	SET 2,(IY+d)				
CBD7	SET 2,A				
CBD0	SET 2,B				
CBD1	SET 2,C				
CBD2	SET 2,D				
CBD3	SET 2,E				
CBD4	SET 2,H				
CBD5	SET 2,L				
CBDE	SET 3,(HL)				
DDCBdDE	SET 3,(IX+d)				
FDCBdDE	SET 3,(IY+d)				
CBDF	SET 3,A				
CBD8	SET 3,B				
CBD9	SET 3,C				
CBDA	SET 3,D				

CÓDIGOS DE LAS INSTRUCCIONES DEL Z80 POR ORDEN ALFABÉTICO

Código objeto	*Instrucción*	*S*	*Z*	*P/V*	*C*
CBDB	SET 3,E				
CBDC	SET 3,H				
CBDD	SET 3,L				
CBE6	SET 4,(HL)				
DDCBdE6	SET 4,(IX+d)				
FDCBdE6	SET 4,(IY+d)				
CBE7	SET 4,A				
CBE0	SET 4,B				
CBE1	SET 4,C				
CBE2	SET 4,D				
CBE3	SET 4,E				
CBE4	SET 4,H				
CBE5	SET 4,L				
CBEE	SET 5,(HL)				
DDCBdEE	SET 5,(IX+d)				
FDCBdEE	SET 5,(IY+d)				
CBEF	SET 5,A				
CBE8	SET 5,B				
CBE9	SET 5,C				
CBEA	SET 5,D				
CBEB	SET 5,E				
CBEC	SET 5,H				
CBED	SET 5,L				
CBF6	SET 6,(HL)				
DDCBdF6	SET 6,(IX+d)				
FDCBdF6	SET 6,(IY+d)				
CBF7	SET 6,A				
CBF0	SET 6,B				
CBF1	SET 6,C				
CBF2	SET 6,D				
CBF3	SET 6,E				
CBF4	SET 6,H				
CBF5	SET 6,L				
CBFE	SET 7,(HL)				
DDCBdFE	SET 7,(IX+d)				
FDCBdFE	SET 7,(IY+d)				
CBFF	SET 7,A				
CBF8	SET 7,B				
CBF9	SET 7,C				
CBFA	SET 7,D				
CBFB	SET 7,E				
CBFC	SET 7,H				
CBFD	SET 7,L				
CB26	SLA (HL)	●	●	●	●
DDCBd26	SLA (IX+d)	●	●	●	●
FDCBd26	SLA (IY+d)	●	●	●	●
CB27	SLA A	●	●	●	●
CB20	SLA B	●	●	●	●

Código objeto	*Instrucción*	*S*	*Z*	*P/V*	*C*
CB21	SLA C	●	●	●	●
CB22	SLA D	●	●	●	●
CB23	SLA E	●	●	●	●
CB24	SLA H	●	●	●	●
CB25	SLA L	●	●	●	●
CB2E	SRA (HL)	●	●	●	●
DDCBd2E	SRA (IX+d)	●	●	●	●
FDCBd2E	SRA (IY+d)	●	●	●	●
CB2F	SRA A	●	●	●	●
CB28	SRA B	●	●	●	●
CB29	SRA C	●	●	●	●
CB2A	SRA D	●	●	●	●
CB2B	SRA E	●	●	●	●
CB2C	SRA H	●	●	●	●
CB2D	SRA L	●	●	●	●
CB3E	SRL (HL)	●	●	●	●
DDCBd3E	SRL (IX+d)	●	●	●	●
FDCBd3E	SRL (IY+d)	●	●	●	●
CB3F	SRL A	●	●	●	●
CB38	SRL B	●	●	●	●
CB39	SRL C	●	●	●	●
CB3A	SRL D	●	●	●	●
CB3B	SRL E	●	●	●	●
CB3C	SRL H	●	●	●	●
CB3D	SRL L	●	●	●	●
96	SUB (HL)	●	●	●	●
DD96d	SUB (IX+d)	●	●	●	●
FD96d	SUB (IY+d)	●	●	●	●
97	SUB A	●	●	●	●
90	SUB B	●	●	●	●
91	SUB C	●	●	●	●
92	SUB D	●	●	●	●
93	SUB E	●	●	●	●
94	SUB H	●	●	●	●
95	SUB L	●	●	●	●
D6d	SUB d	●	●	●	●
AE	XOR (HL)	●	●	●	0
DDAEd	XOR (IX+d)	●	●	●	0
FDAEd	XOR (IY+d)	●	●	●	0
AF	XOR A	●	●	●	0
A8	XOR B	●	●	●	0
A9	XOR C	●	●	●	0
AA	XOR D	●	●	●	0
AB	XOR E	●	●	●	0
AC	XOR H	●	●	●	0
AD	XOR L	●	●	●	0
EEd	XOR d	●	●	●	0

TABLAS DE DESENSAMBLAJE

Instrucciones sin prefijos

– n : byte (8 bits, de 0 a 255);
– nn : doble byte (16 bits, de 0 a 65535);
– d : desplazamiento para el direccionamiento relativo (8 bits).

	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Ø	NOP	LD BC,nn	LD (BC),A	INC BC	INC B	DEC B	LD B,n	RLCA	EX AF,AF'	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DEC C	LD C,n	RRCA
1	DJNZ d	LD DE,nn	LD (DE),A	INC DE	INC D	DEC D	LD D,n	RLA	JR d	ADD HL,DE	LD A,(DE)	DEC DE	INC E	DEC E	LD E,n	RRA
2	JR NZ,d	LD HL,nn	LD (nn),HL	INC HL	INC H	DEC H	LD H,n	DAA	JR Z,d	ADD HL,HL	LD HL,(nn)	DEC HL	INC L	DEC L	LD L,n	CPL
3	JR NC,d	LD SP,nn	LD (nn),A	INC SP	INC (HL)	DEC (HL)	LD (HL),n	SCF C,d	JR C,d	ADD HL,SP	LD A,(nn)	DEC SP	INC A	DEC A	LD A,n	CCF
4	LD B,B	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	LD C,C	LD C,D	LD C,E	LD C,H	LD C,L	LD C,(HL)	LD C,A
5	LD D,B	LD D,C	LD D,D	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	LD E,E	LD E,H	LD E,L	LD E,(HL)	LD E,A
6	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD L,L	LD L,(HL)	LD L,A
7	LD (HL),B	LD (HL),C	LD (HL),D	LD (HL),E	LD (HL),H	LD (HL),L	HALT	LD (HL),A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,(HL)	LD A,A
8	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)	ADD A,A	ADC A,B	ADC A,C	ADC A,D	ADC A,E	ADC A,H	ADC A,L	ADC A,(HL)	ADC A,A
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB (HL)	SUB A	SBC A,B	SBC A,C	SBC A,D	SBC A,E	SBC A,H	SBC A,L	SBC A,(HL)	SBC A,A
A	AND B	AND C	AND D	AND E	AND H	AND L	AND (HL)	AND A	XOR B	XOR C	XOR D	XOR E	XOR H	XOR L	XOR (HL)	XOR A
B	OR B	OR C	OR D	OR E	OR H	OR L	OR (HL)	OR A	CP B	CP C	CP D	CP E	CP H	CP L	CP (HL)	CP A
C	RET NZ	POP BC	JP NZ,nn	JP nn	CALL NZ,nn	PUSH BC	ADD A,n	RST Ø	RET Z	RET	JP Z,nn		CALL Z,nn	CALL nn	ADC A,n	RST 8
D	RET NC	POP DE	JP NC,nn	OUT (n),A	CALL NC,nn	PUSH DE	SUB n	RST 16	RET C	EXX	JP C,nn	IN A,(n)	CALL C,nn		SBC A,n	RST 24
E	RET PO	POP HL	JP PO,nn	EX (SP),HL	CALL PO,nn	PUSH HL	AND n	RST 32	RET PE	JP (HL)	JP PE,nn	EX DE,HL	CALL PE,nn		XOR n	RST 40
F	RET P	POP AF	JP P,nn	DI	CALL P,nn	PUSH AF	OR n	RST 48	RET M	LD SP,HL	JP M,nn	EI	CALL M,nn		CP n	RST 56

Instrucciones con el prefijo CB

Todas las instrucciones de esta tabla deben ir precedidas por el prefijo CB.

	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Ø	RLC B	RLC C	RLC D	RLC E	RLC H	RLC L	RLC (HL)	RLC A	RRC B	RRC C	RRC D	RRC E	RRC H	RRC L	RRC (HL)	RRC A
1	RL B	RL C	RL D	RL E	RL H	RL L	RL (HL)	RL A	RR B	RR C	RR D	RR E	RR H	RR L	RR (HL)	RR A
2	SLA B	SLA C	SLA D	SLA E	SLA H	SLA L	SLA (HL)	SLA A	SRA B	SRA C	SRA D	SRA E	SRA H	SRA L	SRA (HL)	SRA A
3									SRL B	SRL C	SRL D	SRL E	SRL H	SRL L	SRL (HL)	SRL A
4	BIT Ø,B	BIT Ø,C	BIT Ø,D	BIT Ø,E	BIT Ø,H	BIT Ø,L	BIT Ø,(HL)	BIT Ø,A	BIT 1,B	BIT 1,C	BIT 1,D	BIT 1,E	BIT 1,H	BIT 1,L	BIT 1,(HL)	BIT 1,A
5	BIT 2,B	BIT 2,C	BIT 2,D	BIT 2,E	BIT 2,H	BIT 2,L	BIT 2,(HL)	BIT 2,A	BIT 3,B	BIT 3,C	BIT 3,D	BIT 3,E	BIT 3,H	BIT 3,L	BIT 3,(HL)	BIT 3,A
6	BIT 4,B	BIT 4,C	BIT 4,D	BIT 4,E	BIT 4,H	BIT 4,L	BIT 4,(HL)	BIT 4,A	BIT 5,B	BIT 5,C	BIT 5,D	BIT 5,E	BIT 5,H	BIT 5,L	BIT 5,(HL)	BIT 5,A
7	BIT 6,B	BIT 6,C	BIT 6,D	BIT 6,E	BIT 6,H	BIT 6,L	BIT 6,(HL)	BIT 6,A	BIT 7,B	BIT 7,C	BIT 7,D	BIT 7,E	BIT 7,H	BIT 7,L	BIT 7,(HL)	BIT 7,A
8	RES Ø,B	RES Ø,C	RES Ø,D	RES Ø,E	RES Ø,H	RES Ø,L	RES Ø,(HL)	RES Ø,A	RES 1,B	RES 1,C	RES 1,D	RES 1,E	RES 1,H	RES 1,L	RES 1,(HL)	RES 1,A
9	RES 2,B	RES 2,C	RES 2,D	RES 2,E	RES 2,H	RES 2,L	RES 2,(HL)	RES 2,A	RES 3,B	RES 3,C	RES 3,D	RES 3,E	RES 3,H	RES 3,L	RES 3,(HL)	RES 3,A
A	RES 4,B	RES 4,C	RES 4,D	RES 4,E	RES 4,H	RES 4,L	RES 4,(HL)	RES 4,A	RES 5,B	RES 5,C	RES 5,D	RES 5,E	RES 5,H	RES 5,L	RES 5,(HL)	RES 5,A
B	RES 6,B	RES 6,C	RES 6,D	RES 6,E	RES 6,H	RES 6,L	RES 6,(HL)	RES 6,A	RES 7,B	RES 7,C	RES 7,D	RES 7,E	RES 7,H	RES 7,L	RES 7,(HL)	RES 7,A
C	SET Ø,B	SET Ø,C	SET Ø,D	SET Ø,E	SET Ø,H	SET Ø,L	SET Ø,(HL)	SET Ø,A	SET 1,B	SET 1,C	SET 1,D	SET 1,E	SET 1,H	SET 1,L	SET 1,(HL)	SET 1,A
D	SET 2,B	SET 2,C	SET 2,D	SET 2,E	SET 2,H	SET 2,L	SET 2,(HL)	SET 2,A	SET 3,B	SET 3,C	SET 3,D	SET 3,E	SET 3,H	SET 3,L	SET 3,(HL)	SET 3,A
E	SET 4,B	SET 4,C	SET 4,D	SET 4,E	SET 4,H	SET 4,L	SET 4,(HL)	SET 4,A	SET 5,B	SET 5,C	SET 5,D	SET 5,E	SET 5,H	SET 5,L	SET 5,(HL)	SET 5,A
F	SET 6,B	SET 6,C	SET 6,D	SET 6,E	SET 6,H	SET 6,L	SET 6,(HL)	SET 6,A	SET 7,B	SET 7,C	SET 7,D	SET 7,E	SET 7,H	SET 7,L	SET 7,(HL)	SET 7,A

TABLAS DE DESENSAMBLAJE

Instrucciones con el prefijo ED

Todas las instrucciones de esta tabla deben ir precedidas por el prefijo ED.

	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Ø																
1																
2																
3																
4	IN B,(C)	OUT (C),B	SBC HL,BC	LD (nn),BC	NEG	RETN	IM Ø	LD I,A	IN C,(C)	OUT (C),C	ADC HL,BC	LD BC,(nn)		RETI		LD R,A
5	IN D,(C)	OUT (C),D	SBC HL,DE	LD (nn),DE			IM 1	LD A,I	IN E,(C)	OUT (C),E	ADC HL,DE	LD DE,(nn)			IM 2	LD A,R
6	IN H,(C)	OUT (C),H	SBC HL,HL	LD (nn),HL				RRD	IN L,(C)	OUT (C),L	ADC HL,HL	LD HL,(nn)				RLD
7	IN F,(C)		SBC HL,SP	LD (nn),SP					IN A,(C)	OUT (C),A	ADC HL,SP	LD SP,(nn)				
8																
9																
A	LDI	CPI	INI	OUTI					LDD	CPD	IND	OUTD				
B	LDIR	CPIR	INIR	OTIR					LDDR	CPDR	INDR	OTDR				
C																
D																
E																
F																

Instrucciones indexadas

Todas las instrucciones de esta tabla deben ir precedidas por el prefijo DD, para el caso del registro de índice IX, y por FD para el registro de índice IY.

Código	*Mnemónico*	*Código*	*Mnemónico*
09	ADD IX,BC	CB d 0E	RRC (IX+d)
19	ADD IX,DE	CB d 16	RL (IX+d)
21	LD IX,nn	CB d 1E	RR (IX+d)
22	LD (nn),IX	CB d 26	SLA (IX+d)
23	INC IX	CB d 2E	SRA (IX+d)
29	ADD IX,IX	CB d 3E	SRL (IX+d)
2A	LD IX,(nn)	CB d 46	BIT 0,(IX+d)
2B	DEC IX	CB d 4E	BIT 1,(IX+d)
34	INC (IX+d)	CB d 56	BIT 2,(IX+d)
35	DEC (IX+d)	CB d 5E	BIT 3,(IX+d)
36	LD (IX+d),nn	CB d 66	BIT 4,(IX+d)
39	ADD IX,SP	CB d 6E	BIT 5,(IX+d)
46	LD B,(IX+d)	CB d 76	BIT 6,(IX+d)
4E	LD C,(IX+d)	CB d 7E	BIT 7,(IX+d)
56	LD D,(IX+d)	CB d 86	RES 0,(IX+d)
5E	LD E,(IX+d)	CB d 8E	RES 1,(IX+d)
66	LD H,(IX+d)	CB d 96	RES 2,(IX+d)
6E	LD L,(IX+d)	CB d 9E	RES 3,(IX+d)
70	LD (IX+d),B	CB d A6	RES 4,(IX+d)
71	LD (IX+d),C	CB d AE	RES 5,(IX+d)
72	LD (IX+d),D	CB d B6	RES 6,(IX+d)
73	LD (IX+d),E	CB d BE	RES 7,(IX+d)
74	LD (IX+d),H	CB d C6	SET 0,(IX+d)
75	LD (IX+d),L	CB d CE	SET 1,(IX+d)
77	LD (IX+d),A	CB d D6	SET 2,(IX+d)
7E	LD 1,(IX+d)	CB d DE	SET 3,(IX+d)
86	ADD A,(IX+d)	CB d E6	SET 4,(IX+d)
8E	ADC A,(IX+d)	CB d EE	SET 5,(IX+d)
96	SUB (IX+d)	CB d F6	SET 6,(IX+d)
9E	SBC A,(IX+d)	CB d FE	SET 7,(IX+d)
A6	AND (IX+d)	E1	POP IX
AE	XOR (IX+d)	E3	EX (SP),IX
B6	OR (IX+d)	E5	PUSH IX
BE	CP (IX+d)	E9	JP (IX)
CB d 06	RLC (IX+d)	F9	LD SP,IX

GENERALIDADES

El soft interno del Amstrad puede dividirse en tres partes:

- la ROM inferior: contiene los gestores que se mencionan más abajo, las rutinas matemáticas y el generador de caracteres;
- la ROM superior: contiene el Basic propiamente dicho;
- la zona de trabajo en memoria RAM: contiene las variables del sistema, los vectores de llamadas a las rutinas de la ROM inferior y diferentes memorias tampones (buffers) utilizadas por los gestores y el Basic.

Los gestores pueden dividirse en ocho grandes clases

El gestor de teclado

Se ocupa de la gestión del teclado, de generar los caracteres de las teclas de función, de comprobar el BREAK y de gestionar las manecillas de juegos.

El gestor del modo texto

Se ocupa de la gestión del cursor, de los códigos de control y de la visualización de los caracteres en la pantalla.

El gestor gráfico

Se ocupa de trazar los puntos y tirar las líneas en la pantalla.

El gestor de pantalla

Interrelaciona el texto y los gráficos con los circuitos especializados de gestión de pantalla.

El gestor de cassette

Se ocupa de la lectura, de la escritura y del control del motor del cassette.

El gestor sonoro

Se ocupa de la gestión de las colas sonoras, de las envolventes, del sincronismo, etc.

El núcleo (kernel)

Es el corazón del sistema operativo que se ocupa de las interrupciones, del arranque de los programas y de la selección de las memorias ROM.

Los interfaces con el hardware

Se ocupa de la gestión de la impresora y de la relación (interfaces) con el hard al nivel más elemental.

El bloque de salto

Se ocupa de reinicializar todos los vectores.

Para más facilidad, el soft del sistema se presentará de la siguiente manera:

- La tabla de los puntos de entrada en memoria RAM de las rutinas del sistema utilizadas por los gestores (página 81).
- Los vectores de indirección en memoria RAM (página 105).
- Los vectores del núcleo y los restart en memoria RAM (página 107).
- Los vectores de llamada a las rutinas matemáticas en memoria RAM (página 111).
- Las principales variables del sistema en memoria RAM (página 115).
- Las direcciones principales de la ROM inferior (página 120).
- Las direcciones principales de la ROM superior (página 126).
- La tabla de correspondencia entre los vectores y las direcciones de la ROM inferior (página 131).
- La tabla de los puntos de entrada de las palabras-clave del Basic (página 133).
- Los diferentes bloques de control del sistema (página 135).

Siguiendo al número de punto de entrada, se ha colocado la dirección hexadecimal de cada una de las subrutinas, junto con una breve explicación. Las abreviaturas **CE** y **CS** significan **Condiciones de Entrada** y **Condiciones de Salida**.

El gestor de teclado

00 BB00 Inicialización del gestor de teclado.
No hay CE.
CS : AF BC, DE y HL se modifican. Todos los demás registros permanecen inalterados.

01 BB03 RESET del gestor de teclado.
No hay CE.
CS : AF, BC, DE y HL se modifican. Todos los demás registros permanecen inalterados.

02 BB06 El gestor de teclado espera el carácter escrito en teclado.
No hay CE.
CS : si el bit de CARRY es verdadero, el Acumulador contiene el carácter que se ha tecleado. Todos los registros permanecen inalterados.

03 BB09 Lectura de un carácter.
Esta rutina comprueba si hay un carácter disponible desde el teclado.
No hay CE.
CS : si está disponible un carácter, el bit de CARRY es verdadero y A contiene el carácter. Si no hay ningún carácter disponible, el bit de CARRY es falso y A se modifica. Todos los demás registros permanecen inalterados.

04 BB0C Reserva un carácter para la próxima llamada de la rutina precedente.
CE : A contiene el carácter que se debe guardar.
CS : todos los registros permanecen inalterados.

05 BB0F Posiciona una cadena de caracteres asociada a un código.
CE : B contiene el código que debe ser asociado a una cadena de caracteres.
C contiene la longitud de la cadena.
HL contiene su dirección.
CS : si hay expansión, el bit de CARRY es verdadero. Si la cadena es demasiado larga o el código no es válido, el bit de CARRY es falso. A, BC, DE y HL se modifican.

06 BB12 Lee un carácter desde una cadena de caracteres expandida. Los caracteres contenidos en la cadena están numerados partiendo de 0.
CE : A contiene el código de expansión. L contiene el número del carácter.

CS : si se ha encontrado el carácter, A contiene el carácter y el bit de CARRY es verdadero. Si el código es incorrecto o si la cadena de caracteres no es demasiado larga, el bit de CARRY es falso y A se modifica. DE se modifica.

07 BB15 Disposición de un tampón para una cadena de caracteres expandida.

CE : DE contiene la dirección del tampón y HL su longitud.

CS : si todo es correcto, el bit de CARRY es verdadero; de lo contrario es falso. Los registros A, BC, DE y HL se modifican.

08 BB18 Espera de un carácter proveniente del teclado.
No hay CE.

CS : el bit de CARRY es verdadero y A contiene el carácter que ha sido tecleado. Todos los registros permanecen inalterados.

09 BB1B Comprueba si está disponible una tecla proveniente del teclado.
No hay CE.

CS : si está disponible una tecla, el bit de CARRY es verdadero y A contiene el carácter; de lo contrario CARRY es falso.

10 BB1E Comprueba si se ha pulsado una tecla.
Permite comprobar también la manecilla de juegos.

CE : A contiene el número de la tecla que debe ser comprobada.

CS : si no se ha pulsado la tecla, el bit de CARRY es verdadero; si se ha pulsado, el bit de CARRY es falso y A y HL se modifican y C contiene el estado de las teclas SHIFT y CONTROL.

11 BB21 Verifica si la tecla de fijación de las mayúsculas está pulsada (CAPS LOCK).
No hay CE.

CS : L contiene el estado de la tecla SHIFT y A contiene el estado de la tecla de fijación de las mayúsculas; A contiene 00 si dicha tecla no está pulsada y FF si lo está.
El registro AF se modifica.

12 BB24 Lectura del estado de la manecilla de juegos.
No hay CE.

CS : H contiene el estado de la manecilla de juegos n.º 0.
L contiene el estado de la manecilla de juegos n.º 1.
A contiene el estado de la manecilla de juegos n.º 0.
La correspondencia de los bits es la misma que la de la función JOY ya descrita en las funciones Basic.

13 BB27 Posiciona el código que se proporcionará cuando se pulse una tecla sin CTRL ni SHIFT.

CE : A contiene el número de la tecla.
B contiene el código ASCII que proporcionará dicha tecla.

CS : AF y HL se modifican.

14 BB2A Proporciona el código correspondiente a un número de tecla pulsada.

CE : A contiene el número de la tecla.

CS : A contiene el código ASCII correspondiente a la tecla.
HL y F se modifican.

15 BB2D Posiciona el código que se proporcionará cuando se pulse una tecla conjuntamente con la de SHIFT.

CE : A contiene el número de la tecla.
B contiene el código ASCII que proporcionará dicha tecla.

CS : AL y HL se modifican.

16 BB30 Proporciona el código correspondiente a una tecla pulsada conjuntamente con la tecla SHIFT.

CE : A contiene el número de la tecla.

CS : A contiene el código ASCII correspondiente a la tecla.
HL se modifica.

17 BB33 Posiciona el código que proporcionará una tecla pulsada conjuntamente con la tecla CTRL.

CE : A contiene el número de la tecla.
B contiene el código ASCII que dicha tecla proporcionará.

CS : AF y HL se modifican.

18 BB36 Proporciona el código ASCII correspondiente a la tecla pulsada conjuntamente con la tecla CTRL.

CE : A contiene el número de la tecla.

CS : A contiene el código ASCII correspondiente a la tecla.
HL se modifica.

19 BB39 Posiciona la entrada en la tabla de las teclas que deben repetirse.

CE : A contiene el número de la tecla.
Si la tecla puede repetirse, B contiene FF; de lo contrario B contiene 00.

CS : AF, BC y HL se modifican.

20 BB3C Comprueba si una tecla cuyo número se proporciona puede repetirse.

CE : A contiene el número de la tecla.

CS : si la tecla puede repetirse, el bit 0 es falso; si no puede repetirse, el bit 0 es verdadero. De todas maneras, el bit de CARRY es falso y AF y HL se modifican.

21 BB3F Posiciona el tiempo de retardo antes de la primera repetición, así como el tiempo previsto entre dos repeticiones.

CE : H contiene el retardo antes de la primera repetición.
L contiene la velocidad de repetición. Estos tiempos se expresan en cincuentésimas de segundo.

CS : AF se modifica.

22 BB42 Lectura de la velocidad de repetición y del tiempo previsto antes de la primera repetición.
No hay CE.
CS : H contiene el retardo antes de la primera repetición (en 1/50.º de segundo) y L contiene la velocidad de repetición. AF se modifica.

23 BB45 Posiciona el mecanismo del BREAK.
CE : DE contiene la dirección de la rutina de tratamiento del BREAK. C contiene la dirección de la ROM seleccionada para esta rutina.
CS : AF, BC, DE y HL se modifican.
Nota: este mecanismo puede desconectarse por la llamada a la rutina siguiente.

24 BB48 Desconexión del mecanismo del BREAK.
No hay CE.
CS : AF y HL se modifican.

25 BB4B Genera una interrupción de BREAK si se ha establecido BREAK por medio de la rutina 23.
No hay CE.
CS : AF y HL se modifican.

El gestor del modo texto

26 BB4E Inicialización del modo TEXTO.
No hay CE.
CS : AF, BC, DE y HL se modifican.

27 BB51 RESET del modo TEXTO.
No hay CE.
CS : AF, BC, DE y HL se modifican.

28 BB54 Permite que un carácter se coloque en la pantalla en modo texto.
No hay CE.
CS : AF se modifica.

29 BB57 Prohibición de colocar caracteres en la pantalla.
No hay CE.
CS : AF se modifica.

30 BB5A Salida en pantalla en modo texto de un carácter o de un código de control interpretado (los códigos de control son todos caracteres comprendidos entre 0 y 1F).
CE : A contiene el carácter que hay que enviar.
CS : todos los registros permanecen inalterados.

31 BB5D Salida en pantalla en modo texto de un carácter o de un gráfico correspondiente a un código de control.

CE : A contiene el carácter que se debe imprimir.

CS : AF, BC, DE y HL se modifican.

32 BB60 Lectura de un carácter proveniente de la pantalla en la posición actual del cursor.
No hay CE.

CS : si ha sido reconocido un carácter, el bit de CARRY es verdadero y A contiene este carácter. En caso contrario, el bit de CARRY es falso y A contiene 0.

33 BB63 Posicionamiento ON u OFF del dispositivo de tratamiento de caracteres gráficos.

CE : A=0 si está impedida la escritura de gráficos (OFF). En caso contrario (ON), A es distinto de 0.

CS : AF se modifica.

34 BB66 Posiciona el tamaño de la ventana de texto actual.

CE : H contiene la columna de los dos primeros ángulos.
D contiene la columna de los otros dos ángulos.
L contiene la línea de dos ángulos.
E contiene la línea de los otros dos ángulos.

CS : AF, BC, DE y HL se modifican.

35 BB69 Lectura del tamaño de la ventana actual.
No hay CE.

CS : si la ventana ocupa la pantalla entera, el CARRY es falso; de lo contrario es verdadero. En ambos casos, H contiene el número de la columna izquierda, D el número de la columna derecha, L el número de la línea de arriba y E el número de la línea de abajo. A se modifica.

36 BB6C Borrado de la ventana actual (CLS).
No hay CE.

CS : AF, BC, DE y HL se modifican.

37 BB6F Determina la posición horizontal del cursor.

CE : A contiene el número de columna del cursor.

CS : AF y HL se modifican.

38 BB72 Determina la posición vertical del cursor.

CE : A contiene el número de línea del cursor.

CS : AF y HL se modifican.

39 BB75 Determina la posición del cursor.

CE : H contiene el número de columna del cursor.
L contiene el número de línea del cursor.

CS : AF y HL se modifican.

TABLA DE LOS PUNTOS DE ENTRADA DE LAS RUTINAS DEL SISTEMA

40 BB78 Lectura de la posición del cursor.
No hay CE.
CS : H contiene el número de columna del cursor.
L contiene el número de línea del cursor.
A contiene el contador de desplazamiento (scrolling).

41 BB7B Utilización de la visualización del cursor en modo texto.
No hay CE.
CS : AF se modifica.

42 BB7E Inutilización de la visualización del cursor en modo texto.
No hay CE.
CS : AF se modifica.

43 BB81 Autoriza la visualización del cursor para el sistema.
No hay CE.
No hay CS.

44 BB84 Impide la visualización del cursor para el sistema.
No hay CE.
No hay CS.

45 BB87 Comprueba si la posición del cursor se halla en el interior de la ventana.
CE : H contiene el número de columna de la posición que hay que comprobar.
L contiene el número de línea de la posición que hay que comprobar.
CS : H contiene el número de columna en la que se imprimirá el carácter.
L contiene el número de línea en la que se imprimirá el carácter.
A y F se modifican. Si la impresión no debe producir desplazamiento de la ventana (scrolling), el bit de CARRY es verdadero y B se modifica. Si debe producir el desplazamiento de la ventana, el bit de CARRY es falso y B contiene FF. Si debe producir un desplazamiento inverso, CARRY es falso y B contiene 00.

46 BB8A Posiciona el cursor en la pantalla.
No hay CE.
CS : AF se modifica.

47 BB8D Suprime el cursor de la pantalla.
No hay CE.
CS : AF se modifica.

48 BB90 Determina el color de los caracteres.
CE : A contiene el número de tinta.
CS : AF y HL se modifican.

49 BB93 Lectura del color de los caracteres.
No hay CE.
CS : A contiene el número de tinta.
F se modifica.

50 BB96 Determina el color de fondo para texto (papel).
CE : A contiene el número de tinta.
CS : AF y HL se modifican.

51 BB99 Lectura del color de fondo para texto (papel).
CE : No hay CE.
CS : A contiene el número de color del papel.
A y F se modifican.

52 BB9C Invierte el color de los caracteres y el color de fondo.
No hay CE.
CS : AF y HL se modifican.

53 BB9F Permite o impide la visualización del fondo.
CE : si aparece el fondo (modo opaco), A=0; si no aparece el fondo (modo transparente), A es distinto de 0.
CS : AF y HL se modifican.

54 BBA2 Comprueba si puede verse el fondo o no.
No hay CE.
CS : A=0 si el fondo puede verse (modo opaco); de lo contrario (modo transparente), A es distinto de 0. DE, HL y F se modifican.

55 BBA5 Lectura de la dirección de una matriz de caracteres.
CE : A contiene el carácter que hay que buscar en la matriz.
CS : A y F se modifican. Si la matriz es una matriz definida por el usuario, CARRY es verdadero. Si la matriz está en la ROM, CARRY es falso y HL contiene la dirección de la matriz.

56 BBA8 Posicionamiento de una matriz para un carácter definido por el usuario.
CE : A contiene el carácter en el que debe posicionarse la matriz, y HL contiene la dirección de la matriz.
CS : si el carácter es definible por el usuario, CARRY es verdadero, de lo contrario es falso. AF, BC, DE y HL se modifican.

57 BBAB Posicionamiento de la dirección de la tabla de una matriz definida por el usuario.
CE : DE contiene el primer carácter de la tabla y HL contiene la dirección de comienzo de la nueva tabla.
CS : si todavía no había tabla, CARRY es falso y A y HL se modifican. Si ya había una tabla definida por el usuario, CARRY es verdadero, A contiene el primer carácter de la antigua tabla, HL contiene la dirección de la antigua tabla y BC y DE se modifican.

58 BBAE Lectura de la dirección de la tabla para una matriz definida por el usuario.
No hay CE.

CS : si no hay tablas de matrices definidas por el usuario, CARRY es falso y A y HL se modifican. Si hay una, CARRY es verdadero, A contiene el primer carácter de la tabla y HL contiene la dirección de comienzo de la tabla.

59 BBB1 Lectura de la dirección de la tabla de los códigos de control.
No hay CE.

CS : HL contiene la dirección de los códigos de control. Todos los demás registros permanecen inalterados.

60 BBB4 Posiciona una nueva tabla de atributo (VDU stream).

CE : A contiene el número de stream requerido.

CS : A contiene el número del antiguo stream.
HL y F se modifican.

61 BBB7 Intercambia los estados de dos tablas de atributos (STREAMS).

CE : B contiene el número del stream 1.
C contiene el número del stream 2.

CS : AF, BC, DE y HL se modifican.

Nota: el stream está compuesto de:

- un número de tinta;
- un número de papel;
- el cursor de posición;
- los límites de las ventanas.

El gestor gráfico

62 BBBA Inicialización del modo gráfico.
No hay CE.

CS : AF, BC, DE y HL se modifican.

63 BBBD RESET del gestor gráfico.
No hay CE.

CS : AF, BC, DE y HL se modifican.

64 BBC0 Desplazamiento de la posición actual en coordenadas absolutas.

CE : DE contiene la coordenada absoluta X.
HL contiene la coordenada absoluta Y.

CS : AF, BC, DE y HL se modifican.

65 BBC3 Desplazamiento de la posición actual en coordenadas relativas a la posición del cursor.

CE : DE contiene la coordenada relativa X.
HL contiene la coordenada relativa Y.

CS : AF, BC, DE y HL se modifican.

66 BBC6 Lectura de la posición actual del gráfico.
No hay CE.
CS : DE contiene la coordenada X.
HL contiene la coordenada Y.
AF se modifica.

67 BBC9 Posiciona el origen del cursor por omisión.
CE : DE contiene la coordenada X del origen.
HL contiene la coordenada Y del origen.
CS : AF, BC, DE y HL se modifican.

68 BBCC Lectura de las coordenadas del origen.
No hay CE.
CS : DE contiene la coordenada X del origen.
HL contiene la coordenada Y del origen.

69 BBCF Posicionamiento de las puertas derecha e izquierda de una ventana gráfica.
CE : DE contiene la coordenada horizontal de un borde.
HL contiene la coordenada horizontal del otro borde.
CS : AF, BC, DE y HL se modifican.

70 BBD2 Posicionamiento del arriba y abajo de una ventana gráfica.
CE : DE contiene la coordenada Y de uno de los bordes.
HL contiene la coordenada Y del otro borde.
CS : AF, BC, DE y HL se modifican.

71 BBD5 Lectura de los bordes derecho e izquierdo de una ventana gráfica.
No hay CE.
CS : DE contiene la coordenada X del borde izquierdo.
HL contiene la coordenada X del borde derecho.
AF se modifica.

72 BBD8 Lectura de los bordes superior e inferior de la ventana gráfica.
No hay CE.
CS : DE contiene la coordenada Y del borde superior de la ventana.
HL contiene la coordenada Y del borde inferior de la ventana.
AF se modifica.

73 BBDB Borrado de una ventana gráfica.
No hay CE.
CS : AF, BC, DE y HL se modifican.

74 BBDE Posicionamiento del color de escritura de los gráficos.
CE : A contiene el color.
CS : AF se modifica.

TABLA DE LOS PUNTOS DE ENTRADA DE LAS RUTINAS DEL SISTEMA

75 BBE1 Lectura del color de escritura gráfica (tinta).
No hay CE.
CS : A contiene el color de escritura.

76 BBE4 Posicionamiento del color de fondo (papel).
CE : A contiene el número del color.
CS : AF se modifica.

77 BBE7 Lectura del color de fondo (papel).
No hay CE.
CS : A contiene el número del color del papel.

78 BBEA Visualiza un punto en la coordenada absoluta especificada.
CE : DE contiene la coordenada absoluta X.
HL contiene la coordenada absoluta Y.
CS : AF, BC, DE y HL se modifican.

79 BBED Visualiza un punto en la coordenada relativa especificada.
CE : DE contiene la coordenada relativa X.
HL contiene la coordenada relativa Y.
CS : AF, BC, DE y HL se modifican.

80 BBF0 Comprueba un punto de coordenadas absolutas.
CE : DE contiene la coordenada absoluta X.
HL contiene la coordenada absoluta Y.
CS : A contiene el color de la tinta para este punto.
BC, DE y HL se modifican.

81 BBF3 Comprueba un punto de coordenadas relativas.
CE : DE contiene la coordenada relativa X.
HL contiene la coordenada relativa Y.
CS : A contiene el color de la tinta para este punto.
BC, DE y HL se modifican.

82 BBF6 Trazado de una línea en coordenadas absolutas.
CE : DE contiene la coordenada absoluta X del punto de llegada.
HL contiene la coordenada absoluta Y del punto de llegada.
La línea será trazada desde la posición actual a la posición absoluta (X,Y).
CS : AF, BC, DE y HL se modifican.

83 BBF9 Trazado de una línea en coordenadas relativas.
CE : DE contiene la coordenada relativa X del punto de llegada.
HL contiene la coordenada relativa Y del punto de llegada.
La línea será trazada desde la posición actual a la posición relativa (X,Y).
CS : AF, BC, DE y HL se modifican.

84 BBFC Escribe un carácter en la pantalla en la posición gráfica actual.

CE : A contiene el carácter a escribir.
CS : AF, BC, DE y HL se modifican.

El gestor de pantalla

85 BBFF Inicialización principal del gestor de pantalla, los modos, tintas y papeles toman sus valores respectivos por omisión.
CE : no hay.
CS : AF, BC, DE y HL se modifican.

86 BC02 Reinicialización del gestor pantalla.
CE : no hay.
CS : AF, BC, DE y HL se modifican.

87 BC05 Posiciona el OFFSET de partida de la pantalla. Al modificar este valor la pantalla puede enrollarse. (SCROLLING.)
CE : HL contiene el OFFSET deseado.
CS : AF y HL se modifican.

88 BC08 Posiciona el punto de partida en memoria RAM del comienzo de la memoria pantalla.
CE : A contiene el byte más significativo de la dirección de partida.
CS : AF y HL se modifican.

89 BC0B Lectura de la dirección de partida de la memoria pantalla y del OFFSET.
CE : no hay.
CS : A contiene el byte más significativo de la dirección de la memoria pantalla y HL contiene el OFFSET actual. F se modifica.

90 BC0E Posiciona la pantalla en un modo concreto.
CE : A contiene el número del modo.
CS : AF, BC, DE y HL se modifican.

91 BC11 Lectura del modo actual.
CE : no hay.
CS : A contiene el número del modo, CARRY y ZERO están posicionados en función del modo. Modo 0: C=1 Z=0, modo 1: C=0 Z=1, modo 2: C=0 Z=0.

92 BC14 Borrado de la pantalla.
CE : no hay.
CS : AF, BC, DE y HL se modifican.

93 BC17 Lectura del tamaño de la pantalla.
CE : no hay.

SOFTWARE INTERNO

CS : B contiene la última columna física de la pantalla. C contiene la última línea. AF se modifica.

94 BC1A Calcula la dirección real de un carácter del que se da la posición en la pantalla (columna, línea).

CE : H contiene la columna y L contiene la línea.

CS : HL contiene la dirección memoria real, B contiene el tamaño en bytes del carácter en la memoria, AF se modifica.

95 BC1D Cálculo de la dirección real de un punto del que se da su posición en la pantalla.

CE : DE contiene la abscisa del punto (X) y HL contiene la ordenada del punto (Y).

CS : HL contiene la dirección real del punto en memoria, B contiene el número de puntos por byte disminuido en 1, C contiene la máscara para el punto, y AF y DE se modifican.

96 BC20 Cálculo de la dirección real del byte a la derecha de la dirección actual real.

CE : HL contiene la dirección actual.

CS : HL contiene la nueva dirección. AF se modifica.

97 BC23 Como 96 (BC20), pero para el byte a la izquierda.

98 BC26 Como 96 (BC20), pero para la línea siguiente (abajo).

99 BC29 Como 96 (BC20), pero para la línea precedente (arriba).

100 BC2C Conversión del número de tinta de modo que proporcione una máscara, que aplicada al byte que representa los puntos, visualice los puntos de este byte en el color de la tinta.

CE : A contiene el número de tinta.

CS : A contiene la máscara y F se modifica.

101 BC2F Conversión inversa de la precedente.

CE : A contiene la máscara.

CS : A contiene el número de la tinta y F se modifica.

102 BC32 Posiciona los colores de la tinta.

CE : A contiene el número de la tinta.
B contiene el primer color.
C contiene el segundo color.

CS : AF, BC, DE y HL se modifican.

103 BC35 Lectura de los colores de la tinta.

CE : A contiene el número de la tinta.

CS : B contiene el primer color.
C contiene el segundo color.
AF, DE y HL se modifican.

104 BC38 Posiciona los colores de la visualización del marco.

CE : B contiene el primer color.
C contiene el segundo color.

CS : AF, BC, DE y HL se modifican.

105 BC3B Lectura de los colores del marco.

CE : no hay.

CS : B contiene el primer color.
C contiene el segundo color.
AF, DE y HL se modifican.

106 BC3E Posiciona la duración del parpadeo de los colores del marco.

CE : H contiene la duración del primer color.
L contiene la duración del segundo color.

CS : AF y HL se modifican.

107 BC41 Lectura de las duraciones del parpadeo de los colores del marco.

CE : no hay.

CS : H contiene la duración del primer color.
L contiene la duración del segundo color.
AF se modifica.

108 BC44 Relleno de un rectángulo con tinta.

CE : A contiene la máscara correspondiente a la tinta.
H contiene el número de la columna izquierda.
D contiene el número de la columna derecha.
L contiene el número de la línea superior.
E contiene el número de la línea inferior.

CS : AF, BC, DE y HL se modifican.

109 BC47 Posicionamiento de una secuencia de bytes en memoria pantalla en una tinta.

CE : A contiene la máscara correspondiente a la tinta.
HL contiene la dirección de la memoria que corresponde al ángulo superior izquierdo.
D contiene el número de bytes.
E contiene el número de líneas.

CS : AF, BC, DE y HL se modifican.

110 BC4A Inversión de dos colores en un carácter.

CE : B contiene la máscara de un color.
C contiene la máscara del otro color.
H contiene el número de la columna.
L contiene el número de la línea.

CS : AF, BC, DE y HL se modifican.

111 BC4D Desplaza la pantalla entera ocho puntos hacia arriba o hacia abajo.

CE : B=0 para el desplazamiento hacia abajo.
B#0 para el desplazamiento hacia arriba.

CS : AF, BC, DE y HL se modifican.

112 BC50 Desplaza una parte de la pantalla ocho puntos hacia arriba o hacia abajo.

CE : B como antes (111).
A contiene la máscara de la tinta para limpiar la nueva línea.
H contiene el número de la columna izquierda.
D contiene el número de la columna derecha.
L contiene el número de la línea de arriba.
E contiene el número de la línea de abajo.

113 BC53 Conversión de la matriz de un carácter de su forma estándar en una serie de máscaras de puntos en función del modo actual.

CE : HL contiene la dirección de la matriz.
DE contiene la dirección del lugar en el que se encontrará el resultado de la conversión.

CS : AF, BC, DE y HL se modifican.

114 BC56 Conversión inversa a la precedente.

CE : A contiene la máscara de la tinta a convertir.
H contiene la columna del carácter.
L contiene la línea del carácter.
DE contiene la dirección donde se construirá la matriz.

CS : AF, BC, DE y HL se modifican.

115 BC59 Posiciona la pantalla para la utilización del modo gráfico.

CE : A contiene el modo (0 = activación, 1 = OR exclusivo, 2 = AND, 3 = OR).

CS : AF, BC, DE y HL se modifican.

116 BC5C Escritura de un punto en la pantalla sin considerar el modo definido por la rutina precedente (115).

CE : B contiene la máscara de la tinta.
C contiene la máscara del punto.
HL contiene la dirección de la memoria del punto.

CS : AF se modifica.

117 BC5F Trazado de una horizontal.

CE : A contiene la máscara de la tinta.
DE contiene la abscisa de partida.
BC contiene la abscisa de llegada.
HL contiene la ordenada.

CS : AF, BC, DE y HL se modifican.

118 BC62 Trazado de una vertical.

CE : A contiene la máscara de la tinta.

DE contiene la abscisa de la línea.
HL contiene la ordenada de partida.
BC contiene la ordenada de llegada.

CS : AF, BC, DE y HL se modifican.

El gestor de cassette

119 BC65 Inicialización del gestor de cassette.

CE : no hay.

CS : AF, BC, DE y HL se modifican.

120 BC68 Posicionamiento de la velocidad de escritura.

CE : HL contiene la longitud de la mitad de un bit a 0.
A contiene la precompensación a aplicar.

CS : AF y HL se modifican.

121 BC6B Autoriza o impide la visualización de mensajes.

CE : A=0 autoriza. A#0 impide.

CS : AF se modifica.

122 BC6E Puesta en marcha del motor del cassette.

CE : no hay.

CS : si el motor está preparado, CARRY es verdadero; si se ha pulsado ESC, CARRY es falso, A contiene el anterior estado del motor.

123 BC71 Parada del motor.

CE : no hay.

CS : como antes (122).

124 BC74 Reposiciona el motor en su anterior estado.

CE : A contiene el estado anterior del motor.

CS : como antes (122).

125 BC77 Posiciona el buffer para la lectura y lee el primer bloque.

CE : B contiene la longitud del nombre del fichero.
HL contiene la dirección del nombre del fichero.
DE contiene la dirección del buffer (2K).

CS : si es correcto, CARRY es verdadero y ZERO es falso.
HL contiene la dirección del buffer que contiene el encabezamiento, DE contiene la dirección de los datos, BC contiene la longitud del fichero y A contiene el tipo de fichero. Si ya se ha utilizado el STREAM (flujo), CARRY es falso y A, BC, DE y HL se modifican. Si se pulsa ESC, CARRY es falso y ZERO es verdadero. AF, BC, DE y HL se modifican. Finalmente, en todos los casos IX se modifica.

SOFTWARE INTERNO

126 BC7A Cierra el fichero.

CE : no hay.

CS : si es correcto, CARRY es verdadero, si no, CARRY es falso. AF, BC, DE y HL se modifican en ambos casos.

127 BC7D Abandona la lectura y cierra el fichero.

CE : no hay.

CS : AF, BC, DE y HL se modifican.

128 BC80 Lectura de un byte.

CE : no hay.

CS : si es correcto, CARRY es verdadero, ZERO es falso y A contiene el carácter leído. Si se encuentra el fin de fichero (EOF), CARRY es falso, ZERO es falso y A se modifica. Si se pulsa ESC, CARRY es falso, ZERO es verdadero y A se modifica. En todos los casos, IX se modifica.

129 BC83 Lectura de un fichero y escritura en memoria.

CE : HL contiene la dirección de escritura.

CS : como en 128 para los bits de CARRY y ZERO. Además, HL contiene los puntos de entrada si la lectura es correcta. En todos estos casos AF, BC, DE, HL e IX se modifican.

130 BC86 Mete el último carácter leído por la rutina 128 en el buffer de lectura.

CE : no hay.

CS : no hay.

131 BC89 Comprueba si se ha llegado al fin del fichero.

CE : no hay.

CS : si se ha alcanzado el fin del fichero, CARRY es falso y ZERO es falso. Si no se ha alcanzado el fin del fichero, CARRY es verdadero y ZERO es falso. Si el usuario ha pulsado ESC (BREAK), CARRY es falso y ZERO es verdadero. En todos los casos, AF e IX se modifican.

132 BC8C Apertura de un fichero para salida (OPENOUT).

CE : B contiene la longitud del nombre del fichero.
HL contiene la dirección del nombre del fichero.
DE contiene la dirección del buffer de 2K disponible para el fichero.

CS : si el fichero ha sido abierto correctamente, CARRY es verdadero, ZERO es falso y HL contiene la dirección del buffer en el que está el encabezamiento que se escribirá al comienzo de cada bloque de datos. Si el usuario ha pulsado ESC, CARRY es falso y ZERO verdadero. Si ya se ha utilizado el buffer, CARRY es falso y ZERO es falso. De todas maneras, AF, BC, DE, HL e IX se modifican.

133 BC8F Cierre de un fichero en salida (CLOSEOUT).

CE : no hay.

CS : si el cierre es correcto, CARRY es verdadero y ZERO es falso. Si el fichero no había sido abierto, CARRY es falso y ZERO es falso. Si se ha pulsado ESC, CARRY es falso y ZERO es verdadero. De todas maneras, AF, BC, DE, HL e IX se modifican.

134 BC92 Cierre inmediato de un fichero en salida.

CE : no hay.

CS : AF, BC, DE y HL se modifican.

135 BC95 Escritura de un carácter en un fichero de salida.

CE : A contiene el carácter a escribir.

CS : si la escritura es correcta, CARRY es verdadero y ZERO es falso. Si el fichero no había sido abierto, CARRY es falso y ZERO es falso. Si se ha pulsado ESC, CARRY es falso y ZERO es verdadero. De todos modos, AF e IX se modifican.

136 BC98 Escritura directa del contenido de la memoria en un fichero de salida.

CE : HL contiene la dirección de la memoria.
DE contiene el número de bytes a escribir.
BC contiene el punto de entrada.
A contiene el tipo de fichero.

CS : véase el anterior, pero AF, BC, DE, HL e IX se modifican.

137 BC9B Genera el directorio del cassette.

CE : DE contiene la dirección del buffer de 2K disponible.

CS : si la lectura ha tenido lugar correctamente, CARRY es verdadero y ZERO es falso. Si el buffer está ocupado, CARRY es falso y ZERO es falso. Si ha habido un error, CARRY es falso y ZERO verdadero. De todos modos, AF, BC, DE, HL e IX se modifican.

138 BC9E Escribe la grabación en el cassette.

CE : HL contiene la dirección de los datos a escribir.
DE contiene el número de bytes a escribir.
A contiene el carácter de sincronización.

CS : si la grabación transcurre correctamente, CARRY es verdadero. Si no, CARRY es falso y A contiene un código de error. En todos los casos AF, BC, DE, HL e IX se modifican.

139 BCA1 Lee la grabación en el cassette.

CE : HL contiene la dirección en la que se escribirán los datos.
DE contiene el número de bytes a leer.
A contiene el carácter de sincronismo.

CS : si la lectura ha transcurrido correctamente, CARRY es verdadero. Si no, CARRY es falso y A contiene un código de error. En todos los casos, AF, BC, DE, HL e IX se modifican.

140 BCA4 Compara una grabación en el cassette con el contenido de la memoria.

CE : HL contiene la dirección de los datos a comparar.
DE contiene el número de bytes a comparar.
A contiene el carácter de sincronismo.

CS : si la comparación es correcta, CARRY es verdadero; si no, CARRY es falso y A contiene un código de error. En todos los casos, AF, BC, DE, HL é IX se modifican.

El gestor sonoro

141 BCA7 Inicializa el gestor sonoro.

CE : no hay.

CS : AF, BC, DE y HL se modifican.

142 BCAA Añade un sonido a una cola sonora.

CE : HL contiene la dirección del programa sonoro que debe encontrarse dentro de los 32K de RAM central.

CS : si el sonido puede añadirse a la cola sonora, el bit de CARRY es verdadero y HL se modifica. Si todas las colas sonoras están llenas y el sonido sólo puede añadirse a una de ellas, el bit de CARRY es falso y HL queda inalterado. De todos modos, AF, BC, DE e IX se modifican. Los demás registros permanecen inalterados.

143 BCAD Verifica si hay lugar en una cola sonora.

CE : A contiene el número de canal que hay que comprobar.
Vale 0 si se quiere comprobar el canal A.
Vale 1 si se quiere comprobar el canal B.
Vale 2 si se quiere comprobar el canal C.

CS : A contiene el estado del canal comprobado.
F, BC, DE y HL se modifican.

144 BCB0 Prepara la ejecución de una interrupción cuando una cola sonora está vacía.

CE : A contiene el número del canal que ha de estar preparado (0=A, 1=B, 2=C).
HL contiene la dirección del programa de interrupción.

CS : AF, BC, DE y HL se modifican.

145 BCB3 Permite aflojar los sonidos mantenidos en cada canal (ver la siguiente rutina).

CE : A contiene el número del canal a aflojar.
(0=A, 1=B, 2=C).

CS : AF, BC, DE y HL se modifican.

146 BCB6 Detención de todos los sonidos.

CE : no hay.

CS : el bit de CARRY sería verdadero si estuviera activo algún sonido. CARRY será falso si no hay ningún sonido. En cualquier caso, AF, BC y HL se modifican.

147 BCB9 Vuelve a poner en marcha todos los sonidos que estén detenidos por la rutina precedente.

CE : no hay.

CS : AF, BC, DE e IX se modifican.

148 BCBC Constitución de una de las 15 envolventes de amplitud programable.

CE : A contiene el número de envolvente.
HL contiene la dirección de los datos que conciernen a la amplitud.

CS : si se ha establecido una envolvente, CARRY es verdadero, HL contiene la dirección del bloque de datos aumentado en 16, A y BC se modifican. Si el número de envolvente es incorrecto, CARRY es falso, A, B y HL se modifican. De cualquier forma, F y DE se modifican.

149 BCBF Constitución de una de las 15 envolventes de frecuencia programable.

CE : A contiene el número de envolvente.
HL contiene la dirección de datos que conciernen a la frecuencia.

CS : si la envolvente de frecuencia está bien establecida, CARRY es verdadero, HL contiene la dirección del bloque de datos aumentado en 16, y A y BC se modifican. Si el número de envolvente es incorrecto, CARRY es falso, y A, BC y HL permanecen inalterados. De todos modos, F y DE se modifican.

150 BCC2 Proporciona la dirección de la envolvente de amplitud.

CE : A contiene el número de envolvente.

CS : si la envolvente de frecuencia ha sido hallada correctamente, CARRY es verdadero, HL contiene la dirección de la envolvente de amplitud y BC contiene la longitud de la envolvente. Si el número de envolvente es incorrecto, CARRY es falso, HL se modifica y BC queda inalterado. De todas maneras, AF se modifica.

151 BCC5 Proporciona la dirección de una envolvente de tono.

CE : A contiene el número de envolvente.

CS : si la envolvente de frecuencia ha sido hallada correctamente, CARRY es verdadero, HL contiene la dirección de la envolvente de tono y BC contiene la longitud de la envolvente. Si el número de envolvente es incorrecto, CARRY es falso, HL se modifica y BC queda inalterado. De todas maneras, AF se modifica.

El núcleo (kernel)

152 BCC8 Limpia todas las filas de interrupción, los cronómetros...
No hay CE.

CS : B contiene la dirección de selección de la ROM, si hay alguna.
DE contiene el punto de entrada al interior de la ROM.
C contiene la dirección de selección de la ROM para un programa en RAM.
AF y HL se modifican.

153 BCCB Encuentra e inicializa todas las ROMs de segundo plano.

CE : DE contiene la dirección del primer byte utilizable.
HL contiene la dirección del último byte utilizable.

CS : DE contiene la dirección del nuevo primer byte utilizable.
HL contiene la dirección del nuevo último byte utilizable.
AF y BC se modifican.

154 BCCE Inicializa una ROM de segundo plano.

CE : C contiene la dirección de selección de la ROM a inicializar.
DE contiene la dirección del primer byte utilizable.
HL contiene la dirección del último byte utilizable.

CS : DE contiene la dirección del nuevo primer byte utilizable.
HL contiene la dirección del nuevo último byte utilizable.
AF y BC se modifican.

155 BCD1 Introduce un RSX (extensión residente del sistema) en soft interno.

CE : BC contiene la dirección de la tabla de órdenes de RSX.
HL contiene la dirección de cuatro bytes disponibles en RAM para el núcleo.

CS : DE se modifica.

156 BCD4 Busca un RSX en las ROMs para efectuar una orden.

CE : HL contiene la dirección en la que se encuentra el nombre de la orden a buscar.

CS : si se ha encontrado un RSX, CARRY es verdadero, C contiene la dirección de selección de la ROM, y HL contiene la dirección de la rutina. Si no se ha encontrado la orden, CARRY es falso. En todo caso, AF, BC y DE se modifican.

157 BCD7 Inicializa y pone un bloque de suceso en la lista de los bloques a activar cuando hay una interrupción que provenga del CRT.

CE : HL contiene la dirección del bloque de suceso.
B contiene la clase de suceso.
C contiene la dirección de selección de la ROM.
DE contiene la dirección de la rutina de suceso.

CS : AF, DE y HL se modifican.

158 BCDA Pone un bloque de suceso en la lista de bloques a activar cuando hay una interrupción que provenga del CRT.

CE : HL contiene la dirección del bloque de suceso.

CS : AF, DE y HL se modifican.

159 BCDD Suprime un bloque de suceso en la lista de los bloques a activar cuando hay una interrupción que provenga del CRT.

CE : HL contiene la dirección del bloque de suceso.

CS : AF, DE y HL se modifican.

160 BCE0 Inicializa y pone un bloque de suceso en la lista de los bloques a activar cuando hay una interrupción rápida (1/300.º de segundo).

CE : HL contiene la dirección del bloque de suceso.
B contiene la clase de suceso.
C contiene la dirección de selección de la ROM.
DE contiene la dirección de la rutina de suceso.

CS : AF, DE y HL se modifican.

161 BCE3 Pone un bloque de suceso en la lista de bloques a activar cuando hay una interrupción rápida (1/300.º de segundo).

CE : HL contiene la dirección del bloque de suceso.

CS : AF, DE y HL se modifican.

162 BCE6 Suprime un bloque de suceso en la lista de los bloques a activar cuando hay una interrupción rápida (1/300.º de segundo).

CE : HL contiene la dirección del bloque de suceso.

CS : AF, DE y HL se modifican.

163 BCE9 Pone un bloque de suceso en la lista de los bloques a activar cuando hay una interrupción normal (1/50.º de segundo).

CE : HL contiene la dirección del bloque de suceso.
DE contiene el valor inicial del contador.
BC contiene el valor de recarga del contador cuando éste alcanza 0.

CS : AF, BC, DE y HL se modifican.

164 BCEC Suprime un bloque de suceso de la lista de bloques a activar cuando hay una interrupción normal (1/50.º de segundo).

CE : HL contiene la dirección del bloque de suceso.

CS : si se ha encontrado el bloque en la lista, CARRY es verdadero y DE contiene el contador; si no, CARRY es falso. De todos modos, AF, DE y HL se modifican.

165 BCEF Inicializa un bloque de suceso.

CE : HL contiene la dirección del bloque de suceso.
B contiene la clase de suceso.
C contiene la dirección de selección de la ROM.
DE contiene la dirección de la rutina de suceso.

CS : HL contiene la dirección del bloque de suceso aumentado en 7.

166 BCF2 Acciona un bloque de suceso.

CE : HL contiene la dirección del bloque de suceso.

CS : AF, BC, DE y HL se modifican.

167 BCF5 Limpia todas las filas de espera de los sucesos temporizados.
CE : no hay.
CS : AF y HL se modifican.

168 BCF8 Suprime un suceso temporizado fuera de una fila de espera.
CE : HL contiene la dirección del bloque de suceso.
CS : AF, BC, DE y HL se modifican.

169 BCFB Lectura del suceso siguiente en una fila de espera.
CE : no hay.
CS : si hay un suceso a tratar, CARRY es verdadero y HL contiene la dirección del bloque de suceso. A puede contener el código de prioridad del suceso precedente. Si no hay suceso a tratar, CARRY es falso. De todos modos, AF, DE y HL se modifican.

170 BCFE Trata una rutina de suceso.
CE : HL contiene la dirección del bloque de suceso.
CS : AF, BC, DE y HL se modifican.

171 BD01 Termina el tratamiento de un suceso.
CE : HL contiene la dirección del bloque de suceso.
A contiene el código de prioridad del suceso precedente.
CS : AF, BC, DE y HL se modifican.

172 BD04 Impide los sucesos temporizados normales.
No hay CE.
CS : HL se modifica.

173 BD07 Permite los sucesos temporizados normales.
No hay CE.
CS : HL se modifica.

174 BD0A Impide un suceso particular.
CE : HL contiene la dirección del bloque de suceso.
CS : AF se modifica.

175 BD0D Da el tiempo transcurrido en 1/300.º de segundo.
No hay CE.
CS : DEHL contiene el tiempo transcurrido, en cuatro bytes.

Los interfaces con el hardware

176 BD10 Posiciona el contador al valor requerido.
CE : DEHL contiene el valor en cuatro bytes expresado en 1/300.º de segundo.

CS : AF se modifica.

177 BD13 Carga un programa en RAM y lo ejecuta.

CE : HL contiene la dirección de la rutina que hay que llamar para cargar el programa.

CS : ¿Quién puede saber lo que el programa hará?

178 BD16 Lanza un programa a una ROM de segundo plano.

CE : HL contiene el punto de entrada.
C contiene la dirección de selección de la ROM.

CS : indeterminable.

179 BD19 Espera a que CRT genere una señal que indique el comienzo del barrido vertical.
No hay CE.
No hay CS.

180 BD1C Posiciona el modo pantalla.

CE : A contiene el modo (0, 1 o 2).

CS : AF se modifica.

181 BD1F Posiciona el OFFSET de la memoria pantalla.

CE : A contiene la base de la nueva pantalla.
HL contiene el OFFSET.

CS : AF se modifica.

182 BD22 Pone todas las tintas al mismo color para dar el efecto de borrado de pantalla.

CE : DE contiene la dirección del vector de tinta.

CS : AF se modifica.

183 BD25 Posiciona los colores de las tintas y el del marco.

CE : DE contiene la dirección del vector de tinta.

CS : AF se modifica.

184 BD28 Reinicializa el desvío hacia la impresora.
No hay CE.

CS : AF, BC, DE y HL se modifican.

185 BD2B Envía un carácter a la impresora (con posibilidad de retorno si la impresora está ocupada).

CE : A contiene el carácter que se envía.

CS : si se ha enviado el carácter, CARRY es verdadero. Si la impresora queda ocupada por más tiempo, CARRY es falso. De todos modos, AF se modifica.

186 BD2E Comprueba si la impresora está ocupada (BUSY).
No hay CE.

CS : si la impresora está ocupada, CARRY es verdadero, de lo contrario, es falso.

187 BD31 Envía un carácter a la impresora (que no debe estar ocupada).

CE : A contiene el carácter que se envía.

CS : CARRY es verdadero, AF se modifica.

188 BD34 Envía un dato a un registro del PSG.

CE : A contiene el número del registro.
C contiene el dato.

CS : AF y BC se modifican.

El bloque de salto

189 BD37 Reposiciona los bloques de salto estándar.
No hay CE.

CS : AF, BC, DE y HL se modifican.

Con los vectores de indirección el usuario puede interceptar y alterar un cierto número de acciones del soft de sistema sin tener que reescribir completamente todas las rutinas.

Nota: las direcciones que se dan aquí no son puntos de entrada sino llamadas interiores que pueden ser desviadas.

1 BDCD Coloca el carácter del cursor en pantalla.
No hay CE.
CS : AF se modifica.

2 BDD0 Suprime un carácter de la pantalla.
No hay CE.
CS : AF se modifica.

3 BDD3 Escribe un carácter en la pantalla.
CE : A contiene el carácter a escribir.
H contiene el número de columna.
L contiene el número de línea.
CS : AF, BC, DE y HL se modifican.

4 BDD6 Lee un carácter en la pantalla.
CE : H contiene el número de columna.
L contiene el número de línea.
CS : si se ha encontrado el carácter, CARRY es verdadero y A contiene el carácter; si no, CARRY es falso y A contiene 0. De todos modos, AF, BC, DE y HL se modifican.

5 BDD9 Escritura de un carácter o tratamiento de un código de control.
CE : A contiene el carácter o el código de control.
CS : AF, BC, DE y HL se modifican.

6 BDDC Dibuja un punto.
CE : DE contiene la abscisa del punto.
HL contiene la ordenada.
CS : AF, BC, DE y HL se modifican.

7 BDDF Comprueba un punto.
CE : DE contiene la abscisa del punto.
HL contiene la ordenada.
CS : A contiene la tinta del punto especificado.
A, BC, DE y HL se modifican.

8 BDE2 Traza una línea a partir de la posición actual.

CE : DE contiene la abscisa del punto final.
HL contiene la ordenada del punto final.

CS : AF, BC, DE y HL se modifican.

9 BDE5 Lectura de un punto en la memoria de pantalla y decodificación de su tinta.

CE : HL contiene la dirección pantalla del punto.
C contiene la máscara para el punto.

CS : A contiene la tinta decodificada del punto especificado.
AF se modifica.

10 BDE8 Escribe uno o varios puntos en el modo gráfico actual.

CE : HL contiene la dirección pantalla del o de los puntos.
C contiene la máscara para el o para los puntos.
B contiene la tinta codificada.

CS : AF se modifica.

11 BDEB Limpia la pantalla con la tinta 0.
No hay CE.

CS : AF, BC, DE y HL se modifican.

12 BDEE Comprueba la tecla ESC (BREAK).

CE : interrupción prohibida, y C contiene el estado de las teclas CTRL y SHIFT.

CS : AF y HL se modifican.

13 BDF1 Escribe un carácter en la impresora.

CE : A contiene el carácter.

CS : si el carácter se ha escrito correctamente, CARRY es verdadero. Si la impresora sigue ocupada por más tiempo, CARRY es falso. De todos modos, AF y BC se modifican.

A parte de los puntos de entrada principales del soft de sistema, existen algunas rutinas que gestionan la selección y el estado de las ROM. Estas rutinas no pueden ser en ningún caso modificadas por el usuario.

Los vectores de la parte alta de la memoria

1 B900 Selecciona la ROM superior.
No hay CE.

CS : A contiene el estado anterior de la ROM.
AF se modifica.

2 B903 Corta la ROM superior para reseleccionar la RAM.
No hay CE.

CS : A contiene el estado anterior de la ROM.
AF se modifica.

3 B906 Selecciona la ROM inferior.
No hay CE.

CS : A contiene el estado anterior de la ROM.
AF se modifica.

4 B909 Corta la ROM inferior para reseleccionar la RAM.
No hay CE.

CS : A contiene el estado anterior de la ROM.
AF se modifica.

5 B90C Restaura el estado anterior de una ROM.

CE : A contiene el estado anterior de la ROM.

CS : AF se modifica.

6 B90F Selecciona una ROM superior particular.

CE : C contiene la dirección de selección de la ROM requerida.

CS : C contiene la dirección de selección de la ROM precedente.
B contiene el estado de la ROM precedente.
AF se modifica.

7 B912 Pregunta qué ROM está seleccionada.
No hay CE.

CS : A contiene la dirección de selección de la ROM actual.

8 B915 Pregunta la clase y la versión de una ROM.

CE : C contiene la dirección de selección de la ROM que hay que interrogar.

CS : A contiene la clase de la ROM.
H contiene el número de versión.
L contiene un número de marca.
B y F se modifican.

9 B918 Reselecciona la ROM superior anteriormente seleccionada.

CE : C contiene la dirección de selección de la ROM precedente.
B contiene el estado de la ROM precedente.

CS : BC se modifica.

10 B91B Realiza un desplazamiento de bloque con incremento (LDIR) con las dos ROM desconectadas.

CE : BC, DE y HL se programan como para un LDIR normal.

CS : BC, DE, HL y F están en el mismo estado que después de un LDIR normal.

11 B91E Como arriba, pero con decremento (LDDR).

12 B921 Comprueba si se ha producido un suceso con prioridad superior al suceso actual.
No hay CE.

CS : si se ha producido un suceso con prioridad superior, CARRY es verdadero, si no, es falso. AF se modifica.

Los vectores de la parte baja de la memoria

1 0000 RST 0. Reinicialización completa de la máquina como al conectarla a la corriente.
No hay CE.

CS : ¡No se sale!

2 0008 RST 8. Lanzamiento de una rutina en ROM o en RAM inferior. Los dos bytes que siguen al RST contienen la dirección de ejecución y el estado de la ROM superior. Ver el formato en el capítulo 4.11 (página 135).

CE : todos los registros han pasado a la rutina sin quedar afectados.

CS : dependen sólo de la propia rutina.

3 000B Lanzamiento de una rutina en ROM o en RAM inferior.

CE : HL contiene la dirección inferior de la rutina.

CS : dependen sólo de la rutina.

4 000E Salta a la dirección contenida en BC.

CE : BC contiene la dirección.

CS : dependen sólo de la rutina.

5 0010 RST 10. Llamada a una subrutina de una ROM secundaria. Los dos bytes que siguen al RST contienen la dirección de ejecución y la dirección de selección de la ROM. Ver formato en el capítulo 4.11 (página 135).

CE : los registros han pasado a la rutina sin ser modificados, excepto IY.

CS : dependen sólo de la rutina.

6 0013 Llamada a una subrutina de una ROM secundaria. La dirección está contenida en HL.

CE : HL contiene la dirección y los registros han pasado a la rutina sin ser modificados, excepto IY.

CS : dependen sólo de la rutina.

7 0016 Salta a la dirección contenida en DE.

CE : DE contiene la dirección.

CS : dependen sólo de la rutina.

8 0018 RST 18. Llamada a una subrutina en RAM o en ROM. Los dos bytes que siguen contienen la dirección de la subrutina.

CE : todos los registros han pasado a la subrutina, excepto IY.

CS : dependen sólo de la subrutina.

9 001B Llamada a una subrutina en RAM o en ROM con la dirección en HL.

CE : HL contiene la dirección.
C contiene el byte de selección de la ROM o de la RAM.
Todos los registros han pasado a la rutina, excepto IY.

CS : dependen sólo de la rutina.

10 001E Salta a la dirección contenida en HL.

CE : HL contiene la dirección.

CS : dependen sólo de la rutina.

11 0020 RST 20. Carga, en un acumulador, el contenido de la RAM cuya dirección se encuentra en HL, sea cual sea el estado de las ROM.

CE : HL contiene la dirección.

CS : A contiene el valor leído.

12 0023 Llama a una subrutina en RAM o en ROM. HL contiene la dirección en la que se halla la dirección de la subrutina.

CE : HL contiene la dirección en la que se halla la dirección de la subrutina. Todos los registros han pasado a la subrutina, excepto IY.

CS : dependen de la subrutina.

13 0028 RST 28. Salta a una dirección en ROM inferior. Los dos bytes que siguen al RST contienen la dirección.

CE : todos los registros permanecen inalterados.

CS : dependen de la subrutina.

14 0030 RST 30. Reservado al usuario.
El usuario puede hacer lo que quiera con los bytes comprendidos entre el 30 y el 37 ambos incluidos.

15 0038 RST 38. Punto de entrada de las interrupciones generadas por el hardware.
CE : no hay.

CS : todos los registros permanecen inalterados.

16 003B Rutina de tratamiento de las interrupciones exteriores.
CE : no hay.

CS : AF, BC, DE y HL se modifican.

LOS VECTORES DE LLAMADA A LAS RUTINAS MATEMÁTICAS

Las rutinas matemáticas contenidas en la ROM inferior deben ser llamadas frecuentemente desde la ROM Basic para efectuar todas las funciones Basic de cálculo (+, –, *, /, sen, cos,...).

Se ha creado una serie de vectores para facilitar estas llamadas.

Las funciones matemáticas del Basic funcionan usando un acumulador virtual de seis bytes, ubicado desde B0C1 a B0C6. B0C1 contiene el tipo de la variable (2 = entera, 3 = cadena, 5 = real).

Una variable entera está codificada en dos bytes en binario con signo.

Una variable real es más compleja. Se representa en cinco bytes, siguiendo una codificación binaria especial:

– expresar el número en binario;
– contar el número de cifras significativas situadas delante de la coma y añadirle 128 (80H). Así se obtiene el byte 5;
– suprimir el primer bit de la izquierda y convertir los otros siete bits a decimal. Si el número es negativo, añadir 128 (80H). Así se obtiene el byte 4;
– para obtener los bytes 3, 2 y 1, tomar los bits siguientes en grupos de 8 y convertirlos a decimal.

Ejemplo: codificación de la variable real: –2527.

2527 se escribe en binario 1 0011101 1111 (12 cifras).

Byte 5 = 8C porque : 128 + 12 = 140 = 8C.
Byte 4 = 9D porque : tomar los siete bits siguientes: 0011101 = 29 = 1D. El número es negativo, por tanto añadir 128: 29 + 128 = 157 = 9D.
Byte 3 = F0 porque : los ocho bits siguientes son 1111 0000 = 240 = F0.
Byte 2 y byte 1 = 00 porque no hay más bits.

Así pues, –2527 se codifica: 00 00 F0 9D 8C.

Dirección vector	*Dirección real*	*Significado*
BD3D	2E18	Copia los cinco bytes apuntados por DE hacia la zona apuntada por HL y pasa a A el contenido del byte que se encuentra en la dirección HL-1 (tipo de variable).
BD40	2E29	Copia el contenido de A en los cinco bytes apuntados por DE.
BD43	2E55	Conversión del número binario apuntado por HL en un número con el formato del acumulador (5 bytes).
BD46	2E66	Transforma el valor contenido en los cinco bytes apuntados por HL en un entero contenido en HL.
BD49	2E8E	Transforma el valor contenido en los cinco bytes apuntados por HL en un entero contenido en los dos primeros bytes apuntados por HL.

LOS VECTORES DE LLAMADA A LAS RUTINAS MATEMÁTICAS

Dirección vector	*Dirección real*	*Significado*
BD4C	2EA1	Realiza la función FIX.
BD4F	2EAC	Realiza la función INT.
BD52	2EB6	Rutina utilizada por STR$ y PRINT.
BD55	2F1D	Rutina de transformación.
BD58	333F	Adición de dos números reales. HL apunta sobre una zona de cinco bytes que representan un número de formato real (llamada ACCUM1). DE apunta sobre otra zona de cinco bytes (llamada ACCUM2). A la salida de la rutina, HL apunta siempre a ACCUM1, y ACCUM1 contiene el valor ACCUM1+ACCUM2.
BD5B	3337	Sustracción de dos números reales. HL apunta sobre una zona de cinco bytes que representan un número de formato real (llamada ACCUM1). DE apunta sobre otra zona de cinco bytes (llamada ACCUM2). A la salida de la rutina, HL apunta siempre a ACCUM1, y ACCUM1 contiene el valor ACCUM1-ACCUM2.
BD5E	333B	Sustracción de dos números reales. Como arriba, pero ACCUM1 contiene el valor de ACCUM2-ACCUM1.
BD61	4315	Multiplicación de dos números reales. Como arriba, pero ACCUM1 contiene el valor ACCUM1 * ACCUM2.
BD64	349E	División de dos números reales. Como arriba, pero ACCUM1 contiene el valor de ACCUM1 / ACCUM2.
BD67	3578	Añade A al último byte del número apuntado por HL.
BD6A	359A	Comparación de dos números reales: si ACCUM1 > ACCUM2, entonces A=1 si ACCUM1 < ACCUM2, entonces A=255 si ACCUM1 = ACCUM2, entonces A=0.
BD6D	359A	Negación de un número real. HL apunta a ACCUM1, que contiene el valor de −ACCUM1.
BD70	35E8	Comprueba el real contenido en ACCUM1. HL apunta a ACCUM1: si ACCUM1 > 0, entonces A=1 si ACCUM1 < 0, entonces A=255 si ACCUM1 = 0, entonces A=0.

Dirección vector	*Dirección real*	*Significado*
BD73	31AE	Posicionamiento del modo de cálculo de ángulos en grados o en radianes: si A=0, se está en el modo RADIANES si A#0, se está en el modo GRADOS.
BD76	31A3	A la salida, la zona apuntada por HL en la entrada contiene la constante PI.
BD79	310A	Extracción de la raíz cuadrada de un número real. A la entrada, HL apunta a una zona de cinco bytes que contiene un número. A la salida, esta zona contiene la raíz cuadrada del número.
BD7C	310D	Cálculo de la potencia de un número real. HL apunta a ACCUM1 que contiene el número y DE apunta a ACCUM2 que contiene el exponente. A la salida, ACCUM1 contiene el valor de ACCUM1 elevado a ACCUM2.
BD7F	3014	Cálculo del logaritmo neperiano de un número real. HL apunta a ACCUM1 que contiene el número a la entrada. A la salida, ACCUM1 contiene el valor del logaritmo neperiano del número.
BD82	300F	Cálculo del logaritmo en base 10 de un número real. HL apunta a ACCUM1 que contiene el número a la entrada. A la salida, ACCUM1 contiene el valor del logaritmo decimal del número.
BD85	3090	Cálculo del exponencial de un número. HL apunta a ACCUM1, que a la salida contiene el valor del exponencial del número.
BD88	31BC	Cálculo del seno de un ángulo.
BD8B	31B2	Cálculo del coseno de un ángulo.
BD8E	3231	Cálculo de la tangente de un ángulo.
BD91	3241	Cálculo del arcotangente de un ángulo.
BD94	2E5E	Rutina de evaluación.
BD97	2F94	Rutina de carga de B8E4 y B8E6 en la inicialización.
BD9A	2FA1	Rutina utilizada para la generación de números aleatorios.
BD9D	2FB7	Ídem.

LOS VECTORES DE LLAMADA
A LAS RUTINAS MATEMÁTICAS

Dirección vector	*Dirección real*	*Significado*
BDA0	2FE6	Ídem.
BDA3	3708	Manipulación con HL.
BDA6	370E	Mete 0 en B, 0 en E y 2 en C.
BDA9	3715	Manipulación con HL.
BDAC	3728	Adición de dos números enteros. HL=HL+DE. A=FF en caso de desbordamiento.
BDAF	3731	Sustracción de dos números enteros. HL=HL-DE. A=FF en caso de desbordamiento.
BDB2	3730	Sustracción de dos números enteros. HL=DE-HL. A=FF en caso de desbordamiento.
BDB5	3739	Multiplicación de dos números enteros. HL=HL*DE. A=FF en caso de desbordamiento.
BDB8	377A	División de dos números enteros. HL=HL/DE. DE contiene a la salida el resto de la división.
BDBB	3781	Resto de la división de dos números enteros. HL = resto de HL/DE.
BDBE	3750	Manipulación especialmente tenebrosa entre HL y DE.
BDC1	378C	Rutina utilizada en la instrucción PRINT.
BDC4	37E9	Comparación de dos números enteros: si HL > DE, entonces A=1 si HL < DE, entonces A=FF si HL = DE, entonces A=0.
BDC7	37D4	Negación de un número entero. A la salida HL=−(HL).
BDCA	37E0	Comprobación de HL: si HL > 0, entonces A=1 si HL < 0, entonces A=255 si HL = 0, entonces A=0.

Dirección	*Longitud*	*Significado*
AC00	26	26 veces el código C9 (RET).
AC1C	1	Bit: 0 = auto, 1 = no auto.
AC1D	2	Número de la línea actual (utilizado por AUTO).
AC1F	2	Valor del incremento entre dos líneas (AUTO).
AC24	1	Usado por la instrucción WIDTH.
AC26	2	Usado por la instrucción NEXT.
AC2C	2	Usado por la instrucción FOR.
AC2E	2	Usado por la pareja de instrucciones WHILE-WEND.
AC30	11	Usados por la instrucción ON...GOTO.
ACA4	1	Usado por la instrucción EVERY (valor).
ACA5	256	Buffer de entrada teclado.
AD81	2	Número de línea para la instrucción ON ERROR.
ADA6	2	Puntero para la instrucción RESUME.
ADA8	2	Usados para el tratamiento de error.
ADAA	1	Número de error.
ADAB	2	Dirección del último byte ejecutado.
ADAD	2	Dirección para END, STOP y CONT.
ADB1	1	Número de error para la función ON ERROR GOTO.
ADB2	9	Parámetros usados por la instrucción SOUND.
AE0C	26	Tabla de declaración de variables. Compuesta de 26 bytes (1 por cada letra del alfabeto). Cada byte contiene un código que determina el tipo por omisión de cada variable que empiece por dicha letra.
AE2E	2	Dirección de la línea actual para READ-DATA.
AE30	2	Dirección del comienzo de la lectura DATA para RESTORE.
AE34	2	Usado por ON ERROR GOTO.

LAS PRINCIPALES VARIABLES DEL SISTEMA

Dirección	*Longitud*	*Significado*
AE38	1	Bit: 0 = TROFF, 1 = TRON.
AE72	2	Guardado de DE para utilizarlo en la instrucción CALL.
AE74	1	Guardado del acumulador para usarlo en la instrucción CALL.
AE75	2	Guardado de HL para usarlo en la instrucción CALL.
AE77	2	Guardado de SP para usarlo en la instrucción CALL.
AE79	2	Usado por la instrucción ZONE (dirección).
AE7B	2	HIMEM (dirección superior del Basic).
AE7D	2	Usado por la instrucción SYMBOL (dirección).
AE81	2	Dirección del comienzo del programa Basic (por omisión 016F).
AE83	2	Dirección del fin del programa Basic.
AE85	2	Dirección del comienzo de las tablas de variables.
AE87	2	Dirección de la tabla de variables simples.
AE89	2	Dirección de la tabla de variables de tablas (DIM).
B0BA	1	Bit que comprueba si se ha pulsado una tecla (usado por INKEY).
B0C1	1	Tipo del acumulador virtual.
B0C2	5	Cinco bytes usados por el acumulador virtual.
B1C7	1	Byte para la codificación (máscara) de la tinta.
B1C8	1	Modo pantalla (0, 1 o 2).
B1C9	2	OFFSET de la pantalla comprendido entre 0 y 7FF.
B1CB	1	Byte más significativo del comienzo de la memoria real de pantalla.
B1CC	1	A veces contiene un C3 (jump).
B1CD	2	Contiene la dirección para el jump.
B1D7	1	Longitud del primer periodo de parpadeo del marco.
B1D8	1	Longitud del segundo periodo de parpadeo del marco.

Dirección	*Longitud*	*Significado*
B1DA	32	Colores de las tintas (dos bytes por color).
B1FC	1	Usado para enmarcar.
B20C	1	Número del STREAM.
B285	1	Posición línea cursor.
B286	1	Posición columna cursor.
B287	1	Indicador de ventana.
B288	1	Línea inicio ventana actual.
B289	1	Columna inicio ventana actual.
B28A	1	Línea fin ventana actual.
B28B	1	Columna fin ventana actual.
B28D	1	0 = cursor autorizado, 255 = cursor no autorizado.
B28E	1	0 = visualización no autorizada, 255 = visualización autorizada.
B28F	1	Tinta actual para la pluma.
B290	1	Tinta actual para el papel.
B291	1	0 = visualización del fondo permitida, 255 = visualización del fondo no permitida.
B294	2	Primer carácter y estado de la tabla de matrices definida por el usuario.
B296	2	Dirección de la tabla de matrices definida por el usuario.
B2C3	96	Tabla de los códigos de control.
B328	2	Coordenada del origen del eje X.
B32A	2	Coordenada del eje Y.
B32C	2	Coordenada gráfica X.
B32E	2	Coordenada gráfica Y.
B330	2	Coordenada X de un borde de la ventana gráfica.
B332	2	Coordenada X del otro borde de la ventana gráfica.

SOFTWARE INTERNO

Dirección	*Longitud*	*Significado*
B334	2	Coordenada Y de un borde de la ventana gráfica.
B336	2	Coordenada Y del otro borde de la ventana gráfica.
B338	1	Tinta de la pluma gráfica.
B339	1	Tinta del papel gráfico.
B33A	8	Cuatro zonas de dos bytes usados como memoria temporal de cálculo durante el trazado de una línea.
B342	2	Coordenada X del punto final para el trazado de una línea.
B344	2	Coordenada Y del punto final para el trazado de una línea.
B34C	80	Tabla de valores de las teclas sin SHIFT ni CTRL.
B39C	80	Tabla de valores de las teclas con SHIFT.
B3EC	80	Tabla de valores de las teclas con CTRL.
B43C	80	Tabla de repeticiones para cada tecla.
B4DE	2	Usado para examinar (SCANNING) (dirección).
B4E0	1	Guardado temporal del carácter examinado por SCANNING (BB0C).
B4E9	1	Valor de la velocidad de repetición de las teclas.
B4EA	1	Valor del retardo antes de la repetición de una tecla.
B4EB	10	Tabla de SCANNING de las teclas.
B4F1	1	Estado de la manecilla de juegos 1.
B4F4	1	Estado de la manecilla de juegos 2.
B50C	1	Usado para control del BREAK.
B541	2	Dirección de la tabla de las teclas sin SHIFT ni CTRL.
B543	2	Dirección de la tabla de las teclas con SHIFT.
B545	2	Dirección de la tabla de las teclas con CTRL.
B547	2	Dirección de la tabla de repeticiones de las teclas.
B551		Comienzo de la zona de las variables del gestor sonoro.

Dirección	*Longitud*	*Significado*
B60A	240	15 × 16 bytes con los valores de las envolventes de amplitud.
B6FA	240	15 × 16 bytes con los valores de las envolventes tonales.
B800		Comienzo de la zona de las variables del gestor de cassette.
B800	1	Autorizado el PROMPT MESSAGE (indicador) si es 0, no autorizado si ≠ 0.
B802	1	Indicador de apertura de fichero.
B803	2	Dirección del buffer de 2K para el directorio.
B805	2	Dirección del buffer de lectura.
B819	1	Tipo de fichero cassette.
B81A	2	Dirección actual del buffer de lectura.
B81C	2	Emplazamiento de la memoria de datos.
B81F	2	Longitud lógica del fichero.
B847	1	Tipo del STREAM de escritura.
B84A	2	Dirección del buffer de escritura.
B85F	2	Dirección actual del buffer de escritura.
B8CD	1	Carácter de sincronismo.
B8D1	2	Velocidad de escritura o lectura.
B8F7	1	0 = modo RADIANES, 255 = modo GRADOS.

DIRECCIONES PRINCIPALES DE LA ROM INFERIOR

La ROM inferior contiene las rutinas del sistema (comunicación con el equipo físico), las rutinas matemáticas y el generador de caracteres.

Nota: las direcciones que corresponden a las rutinas ya descritas detalladamente en las páginas 81 a 110, van solamente indicadas con el punto de entrada en la RAM correspondiente, seguido de un *.

Por tanto, aconsejamos consultar las páginas 81 a 110 para tener una información más completa.

005C	BCC8	*	
0099	BD0D	*	
00A3	BD10	*	
0163	BCD7	*	
016A	BCDA	*	
0170	BCDD	*	
0176	BCE0	*	
017D	BCE3	*	
0183	BCE6	*	
01B3	BCE9	*	
01C5	BCEC	*	
01D2	BCEF	*	
01E2	BCF2	*	
021A	BCFE	*	
0228	BCF5	*	
0256	BCFB	*	
0277	BD01	*	
0285	BCF8	*	
028E	BD0A	*	
0295	BD04	*	
029B	BD07	*	
02A1	BCD1	*	
02B2	BCD4	*	
0329	BCCB	*	
0332	BCCE	*	
05DC	BD13	*	
060B	BD16	*	
066D	Mensaje	:	64K MICROCOMPUTER (V1)
068A	Mensaje	:	copyright 1984 Amstrad Consumer Electronics PLC and Locomotive Software Ltd
06F4	Mensaje	:	*** program load failed ***
0727	Lista de compatibles Arnold, Amstrad, Orion, Schneider, Awa, Solavox, Saisho, Triumph, Isp		
0776	BD1C	*	
0786	BD22	*	
0799	BD25	*	
07BA	BD19	*	
07C6	BD1F	*	
07E6	BD28	*	

07F2	BD2B	*
07F8	BDF1	*
0807	BD31	*
081B	BD2E	*
0826	BD34	*
0888	BD37	*
0AA0	BBFF	*
0AB1	BC02	*
0ACA	BC0E	*
0AEC	BC11	*
0AF7	BC14	*
0AF7	BDEB	*
0B3C	BC05	*
0B45	BC08	*
0B50	BC0B	*
0B57	BC17	*
0B64	BC1A	*
0BA9	BC1D	*
0BF9	BC20	*
0C05	BC23	*
0C13	BC26	*
0C2D	BC29	*
0C49	BC59	*
0C68	BDE8	*
0C6B	BC5C	*
0C82	BDE5	*
0C86	BC2C	*
0CA0	BC2F	*
0CE4	BC3E	*
0CE8	BC41	*
0CEC	BC32	*
0CF1	BC38	*
0D14	BC35	*
0D19	BC3B	*
0DB3	BC44	*
0DB7	BC47	*
0DDF	BC4A	*
0DFA	BC4D	*
0E3E	BC50	*
0EF3	BC53	*
0F49	BC56	*
0FC4	BC5F	*
102F	BC62	*
1078	BB4E	*
1088	BB51	*
10E8	BBB4	*
1107	BBB7	*
115E	BB6F	*
1169	BB72	*
1174	BB75	*
1180	BB78	*
11CE	BB87	*
120C	BB66	*
1256	BB69	*
1263	BDCD	*
1263	BDD0	*
1268	BB8A	*
1268	BB8D	*
1279	BB81	*
1281	BB84	*
1289	BB7B	*
129A	BB7E	*
12A9	BB90	*
12AE	BB96	*
12BD	BB93	*
12C3	BB99	*
12C9	BB9C	*
12D3	BBA5	*
12F1	BBA8	*
12FD	BBAB	*
132A	BBAE	*
1334	BB5D	*
134A	BDD3	*
137A	BB9F	*
1387	BBA2	*
13A7	BB63	*
13AB	BB60	*
13C0	BDD6	*
1400	BB5A	*
140C	BDD9	*
144B	BB57	*
1451	BB54	*
146B	Tabla de los códigos de control para el terminal (véase página 170), 96 bytes	
14CB	BBB1	*
1540	BB6C	*
15B0	BBBA	*
15DF	BBBD	*
15F1	BBC3	*
15F4	BBC0	*
15FC	BBC6	*
1604	BBC9	*
1612	BBCC	*
1734	BBCF	*

DIRECCIONES PRINCIPALES DE LA ROM INFERIOR

1779	BBD2	*
17A6	BBD5	*
17BC	BBD8	*
17C5	BBDB	*
17F6	BBDE	*
17FD	BBE4	*
1804	BBE1	*
180A	BBE7	*
1810	BBED	*
1813	BBEA	*
1816	BDDC	*
1824	BBF3	*
1827	BBF0	*
182A	BDD0	*
182A	BDDF	*
1836	BBF9	*
1839	BBF6	*
183C	BDE2	*
1945	BBFC	*
19E0	BB00	*
1A1E	BB03	*
1A3C	BB06	*
1A42	BB09	*
1A77	BB0C	*
1A7B	BB15	*
1AB3	Valor por omisión de las teclas extendidas (RUN para CTRL CR)	
1ABD	BB0F	*
1B2E	BB12	*
1B56	BB18	*
1B5C	BB1B	*
1BB3	BB21	*
1C2F	BDEE	*
1C5C	BB24	*
1C6D	BB3F	*
1C69	BB42	*
1C71	BB45	*
1C82	BB48	*
1C90	BB4B	*
1CA6	BB3C	*
1CAB	BB39	*
1CBD	BB1E	*
1D52	BB27	*
1D3E	BB2A	*
1D57	BB2D	*
1D43	BB30	*
1D5C	BB33	*
1D48	BB36	*
1D69	Tabla de valores por omisión de las teclas del teclado	

1E68	BCA7	*	
1ECB	BCB6	*	
1EE6	BCB9	*	
1F9F	BCAA	*	
204A	BCB3	*	
206C	BCAD	*	
2089	BCB0	*	
2338	BCBC	*	
233D	BCBF	*	
2349	BCC2	*	
234E	BCC5	*	
2370	BC65	*	
237F	BC68	*	
238E	BC6B	*	
2392	BC77	*	
23AB	BC8C	*	
23FC	BC7A	*	
2401	BC7D	*	
2415	BC8F	*	
242E	BC92	*	
2435	BC80	*	
245B	BC95	*	
2496	BC89	*	
249A	BC86	*	
24AB	BC83	*	
24EA	BC98	*	
2528	BC9B	*	
27C5	Mensaje	:	press play then any key
27DB	Mensaje	:	error
27E5	Mensaje	:	REC
27E8	Mensaje	:	and
27ED	Mensaje	:	Read
27F3	Mensaje	:	write
27FA	Mensaje	:	Rewind
2800	Mensaje	:	tape
2805	Mensaje	:	found
280D	Mensaje	:	loading
2815	Mensaje	:	saving
281D	Mensaje	:	OK
2820	Mensaje	:	Block
2826	Mensaje	:	Unnamed
282D	Mensaje	:	File
2836	BCA1	*	
283F	BC9E	*	
2851	BCA4	*	
2A4B	BC6E	*	
2A4F	BC71	*	
2A51	BC74	*	

2E18	BD3D *
2E29	BD40 *
2E55	BD43 *
2E5E	BD94 *
2E66	BD46 *
2E8E	BD49 *
2EA1	BD4C *
2EAC	BD4F *
2EB6	BD52 *
2F10	BD55 *
2F53	Tabla de las potencias de 10. 13 entradas de 5 bytes para los valores de 10 a 10E13
2F94	BD97 *
2FA1	BD9A *
2FB7	BD9D *
2FE6	BDA0 *
300F	BD82 * LOG10
3014	BD7F * LOG
3086	Valor codificado de LOG(2) (0,693147181)
308C	Valor codificado de LOG10(2) (0,301029996)
3090	BD85 * EXP
30CC	Constante 0,5 codificada
30FB	Constante 1,44269504
3100	Constante 88,0296919
3105	Constante −88,7228391
310A	BD79 * SQR
310D	BD7C *
31A3	BD76 * PI
31A9	Constante PI (3,14159265)
31AE	BD73 * DEG-RAD
31B2	BD8B * COS
31BC	BD88 * SIN
31EC	Tabla de 6 números codificados en 5 bytes cada uno para calcular los senos y cosenos
321D	Tabla de 4 números codificados en 5 bytes cada uno para calcular los senos y cosenos
3231	BD8E * TAN
3241	BD91 * ATN
3258	Tabla de 11 números codificados en 5 bytes cada uno para calcular el arcotangente
3337	BD5B *
333B	BD5E *
333F	BD58 *
3415	BD61 *
349E	BD64 *
3578	BD67 *
359A	BD6A *
35E8	BD70 *

35F8	BD6D *
3708	BDA3 *
370E	BDA6 *
3715	BDA9 *
3728	BDAC *
3730	BDB2 *
3731	BDAF *
3739	BDB5 *
3750	BDBE *
377A	BDB8 *
3781	BDBB *
378C	BDC1 *
37D4	BDC7 *
37E0	BDCA *
37E9	BDC4 *
3800	Comienzo de la tabla del generador de caracteres (256 × 8 bytes)
3FFF	Fin de la tabla

DIRECCIONES PRINCIPALES DE LA ROM SUPERIOR

La ROM superior contiene todas las rutinas de tratamiento de todas las palabras-clave del Basic.

C002	Inicialización + envío del mensaje: Basic 1.0
C03F	Mensaje: Basic 1.0
C053	Función EDIT
C090	Entrada principal (visualización de: READY)
C0CC	Mensaje: READY
C0DF	AUTO
C12B	NEW
C132	CLEAR
C20A	PAPER
C212	PEN
C221	BORDER
C22A	INK
C24F	MODE
C25A	CLS
C262	VPOS
C276	POS
C2D2	LOCATE
C2E1	WINDOW
C319	TAG
C320	TAGOFF
C337	Envío del mensaje apuntado por HL
C3E3	WIDTH
C417	EOF
C48C	ORIGIN
C4B5	CLG
C4C6	DRAW
C4CB	DRAWR
C4D0	PLOT
C4D5	PLOTR
C4E9	TEST
C4EE	TESTR
C505	MOVE
C50A	MOVER
C529	FOR
C5FB	NEXT
C6C7	IF
C6E8	GOTO
C6ED	GOSUB
C70F	RETURN
C747	WHILE
C776	WEND
C7C3	ON
C8CB	ON BREAK
C8E1	DI
C8E7	EI

C940	ON SQ
C971	AFTER
C979	EVERY
C99F	REMAIN
CA8F	ERROR
CB23	Mensaje: UNDEFINED LINE
CB33	Rutina de envío del mensaje: BREAK in
CB4F	Mensaje: BREAK
CB55	Mensaje: IN
CB5A	STOP
CB65	END
CBC0	CONT
CBF8	ON ERROR
CC03	RESUME
CC5B	Tabla de los mensajes de error
CE66	Fin de la tabla de los mensajes de error
CF81	Tabla de los puntos de entrada de las operaciones aritméticas y lógicas
D0CA	Tabla de los puntos de entrada de las funciones EOF, ERR, HIMEM, INKEY$, PI, RND, TIME, XPOS e YPOS
D0DC	ERR
C0F4	HIMEM
D107	XPOS
D10E	YPOS
D190	Tabla de los puntos de entrada de las funciones
D219	ROUND
D1EA	MIN
D1EE	MAX
D256	OPENOUT
D25F	OPENIN
D298	CLOSEIN
D2A1	CLOSEOUT
D2C0	SOUND
D31E	RELEASE
D329	SQ
D34E	ENV
D385	ENT
D409	INKEY
D423	JOY
D439	KEY DEF
D494	SPEED
D4DB	PI
D4E7	DEG
D4EB	RAD
D4EF	SQR
D4F4	Rutina de elevación a una potencia
D520	EXP
D525	LOG10
D52A	LOG

DIRECCIONES PRINCIPALES DE LA ROM SUPERIOR

D52F	SIN
D534	COS
D539	TAN
D53E	ATN
D543	Mensaje: RANDOM NUMBER SEED ?
D559	RANDOMIZE
D584	RND
D614	DEFSTR
D618	DEFINT
D61C	DEFREAL
D654	LET
D67D	DIM
D9C0	ERASE
DAF8	LINE
DB28	INPUT
DB77	Mensaje: ? redo from start
DCD9	RESTORE
DCEB	READ
DDEZ	TRON
DDEG	TROFF
DE01	Tabla de los puntos de entrada de las palabras-clave del Basic
DEBA	Fin de la tabla
DFDC	Tabla de las palabras-clave que pueden ir seguidas de un número de línea (GOTO, RESTORE, AUTO, EDIT,...)
E0F7	LIST
E2DD	Rutina de posicionamiento en la tabla de las letras para buscar las palabras-clave
E327	Rutina de prueba que verifica si una palabra-clave se halla en la tabla
E354	Tabla de las direcciones para cada una de las 26 letras del alfabeto
E388	Tabla de las palabras-clave con su código
E64A	Fin de la tabla
E728	DELETE
E7DF	RENUM
E8EF	DATA
E8F3	REM
E9BD	RUN
E9F6	LOAD
EA3C	CHAIN
EAA6	MERGE
EC09	SAVE
F158	PEEK
F15F	POKE
F16D	INP
F177	OUT
F17D	WAIT
F1BA	CALL
F1F6	ZONE
F1FD	PRINT

F2C4	PRINT USING
F47B	WRITE
F4EF	MEMORY
F69D	SYMBOL
F834	LOWER$
F839	Rutina de conversión a minúsculas
F842	UPPER$
F8BA	BIN$
F8C4	HEX$
F8EA	DEC$
F91E	STR$
F93C	LEFT$
F943	RIGHT$
F993	MID$
FA0A	LEN
FA10	ASC
FA16	CHR$
FA24	INKEY$
FA36	STRING$
FA57	SPACE$
FA77	VAL
FAA1	INSTR
FC2D	FRE
FCCC	Adición +
FCE1	Sustracción –
FCF5	Multiplicación *
FD12	División /
FD37	División entera\
FD49	Módulo (resto de la división)
FD58	Función AND (Y LÓGICO)
FD63	Función OR (O LÓGICO)
FD6D	Función XOR (O EXCLUSIVO)
FD85	ABS
FDE8	FIX
FDED	INT
FE8D	CINT
FEC2	UNT
FEEC	CREAL
FEF3	Limpieza del acumulador
FF02	SGN
FF0A	Posicionamiento de un entero en el acumulador
FF16	Conversión a número real
FF1D	Mete el tipo de variable en C
FF23	Mete el tipo de variable en A
FF27	Comprueba si el acumulador contiene un puntero de cadena
FF62	Copia el acumulador en la zona apuntada por DE
FF71	Comprueba si es mayúscula
FF7B	Comprueba si es numérica

DIRECCIONES PRINCIPALES DE LA ROM SUPERIOR

FF8A	Conversión en mayúscula
FFAA	Compara A con el contenido de HL
FFB8	Compara HL y DE
FFBE	Compara HL y BC
FFC4	DE = HL − DE
FFCF	HL = HL − DE
FFDA	BC = HL − DE
FFE7	HL = HL − BC
FFF2	LDIR
FFF5	LDDR
FFF8	JP (HL)
FFF9	Vuelta al contenido de BC
FFFB	Vuelta al contenido de DE

Tabla de las direcciones de desvío real de los diferentes vectores

Dirección vector	*Dirección real*	*Dirección vector*	*Dirección real*	*Dirección vector*	*Dirección real*
BBØØ	19EØ	BB81	1279	BCØ2	ØAB1
BBØ3	1A1E	BB84	1281	BCØ5	ØB3C
BBØ6	1A3C	BB87	11CE	BCØ8	ØB45
BBØ9	1A42	BB8A	1268	BCØB	ØB5Ø
BBØC	1A77	BB8D	1268	BCØE	ØACA
BBØF	1ABD	BB9Ø	12A9	BC11	ØAEC
BB12	1B2E	BB93	12BD	BC14	ØAF7
BB15	1A7B	BB96	12AE	BC17	ØB57
BB18	1B56	BB99	12C3	BC1A	ØB64
BB1B	1B5C	BB9C	12C9	BC1D	ØBA9
BB1E	1CBD	BB9F	137A	BC2Ø	ØBF9
BB21	1BB3	BBA2	1387	BC23	ØCØ5
BB24	1C5C	BBA5	12D3	BC26	ØC13
BB27	1D52	BBA8	12F1	BC29	ØC2D
BB2A	1D3E	BBAB	12FD	BC2C	ØC86
BB2D	1D57	BBAE	132A	BC2F	ØCAØ
BB3Ø	1D43	BBB1	14CB	BC32	ØCEC
BB33	1D5C	BBB4	1ØE8	BC35	ØD14
BB36	1D48	BBB7	11Ø7	BC38	ØCF1
BB39	1CAB	BBBA	15BØ	BC3B	ØD19
BB3C	1CA6	BBBD	15DF	BC3E	ØCE4
BB3F	1C6D	BBCØ	15F4	BC41	ØCE8
BB42	1C69	BBC3	15F1	BC44	ØDB3
BB45	1C71	BBC6	15FC	BC47	ØDB7
BB48	1C82	BBC9	16Ø4	BC4A	ØDDF
BB4B	1C9Ø	BBCC	1612	BC4D	ØDFA
BB4E	1Ø78	BBCF	1734	BC5Ø	ØE3E
BB51	1Ø88	BBD2	1779	BC53	ØEF3
BB54	1451	BBD5	17A6	BC56	ØF49
BB57	144B	BBD8	17BC	BC59	ØC49
BB5A	14ØØ	BBDB	17C5	BC5C	ØC6B
BB5D	1334	BBDE	17F6	BC5F	ØFC4
BB6Ø	13AB	BBE1	18Ø4	BC62	1Ø2F
BB63	13A7	BBE4	17FD	BC65	237Ø
BB66	12ØC	BBE7	18ØA	BC68	237F
BB69	1256	BBEA	1813	BC6B	238E
BB6C	154Ø	BBED	181Ø	BC6E	2A4B
BB6F	115E	BBFØ	1827	BC71	2A4F
BB72	1169	BBF3	1824	BC74	2A51
BB75	1174	BBF6	1839	BC77	2392
BB78	118Ø	BBF9	1836	BC7A	23FC
BB7B	1289	BBFC	1945	BC7D	24Ø1
BB7E	129A	BBFF	ØAAØ	BC8Ø	2435

Dirección vector	*Dirección real*	*Dirección vector*	*Dirección real*	*Dirección vector*	*Dirección real*
BC83	24AB	BCE3	Ø17D	B9Ø9	BA54
BC86	249A	BCE6	Ø183	B9ØC	BA72
BC89	2496	BCE9	Ø1B3	B9ØF	BA7E
BC8C	23AB	BCEC	Ø1C5	B912	BAA2
BC8F	2415	BCEF	Ø1D2	B915	BA83
BC92	242E	BCF2	Ø1E2	B918	BA8C
BC95	245B	BCF5	Ø228	B91B	BAA6
BC98	24EA	BCF8	Ø285	B91E	BAAC
BC9B	2528	BCFB	Ø256	BDCD	1263
BC9E	283F	BCFE	Ø21A	BDDØ	1263
BCA1	2836	BDØ1	Ø277	BDD3	134A
BCA4	2851	BDØ4	Ø295	BDD6	13CØ
BCA7	1E68	BDØ7	Ø29B	BDD9	14ØC
BCAA	1F9F	BDØA	Ø28E	BDDC	1816
BCAD	2Ø6C	BDØD	ØØ99	BDDF	182A
BCBØ	2Ø89	BD1Ø	ØØA3	BDE2	183C
BCB3	2Ø4A	BD13	Ø5DC	BDE5	ØC82
BCB6	1ECB	BD16	Ø6ØB	BDE8	ØC68
BCB9	1EE6	BD19	Ø7BA	BDEB	ØAF7
BCBC	2338	BD1C	Ø776	BDEE	1C2F
BCBF	233D	BD1F	Ø7C6	BDF1	Ø7F8
BCC2	2349	BD22	Ø786	8	B982
BCC5	234E	BD25	Ø799	B	B97C
BCC8	ØØ5C	BD28	Ø7E6	1Ø	BA16
BCCB	Ø329	BD2B	Ø7F2	13	BA1Ø
BCCE	Ø332	BD2E	Ø81B	18	B9BF
BCD1	Ø2A1	BD31	Ø8Ø7	1B	B9B1
BCD4	Ø2B2	BD34	Ø826	2Ø	BACB
BCD7	Ø163	BD37	Ø888	23	B9B9
BCDA	Ø16A	B9ØØ	BA5E	28	BA2E
BCDD	Ø17Ø	B9Ø3	BA68	38	B939
BCEØ	Ø176	B9Ø6	BA4A		

DIRECCIONES DE EJECUCIÓN DE LAS PALABRAS-CLAVE DEL BASIC

Palabra-clave	*Dirección*	*Palabra-clave*	*Dirección*
ABS	FD85	**FOR**	C529
AFTER	C971	**FRE**	FC2D
ASC	FA10	**GOSUB**	C6ED
ATN	D53E	**GOTO**	C6E8
AUTO	C0DF	**HEX$**	F8C4
BIN$	F8BA	**HIMEM**	D0F4
BORDER	C221	**IF**	C6C7
CALL	F1BA	**INSTR**	FAA1
CAT	D246	**INK**	C22A
CHAIN	EA3C	**INKEY**	D409
CHR$	FA16	**INKEY$**	FA24
CINT	FE8D	**INP**	F16D
CLEAR	C132	**INPUT**	DB2B
CLG	C4B5	**INT**	FDED
CLOSEIN	D298	**JOY**	D423
CLOSEOUT	D2A1	**KEY**	D439
CLS	C25A	**LEFT$**	F93C
CONT	CBC0	**LEN**	FA0A
COS	D534	**LET**	D654
CREAL	FEEC	**LINE**	DAF8
DATA	E8EF	**LIST**	E0F7
DEC$	F8EA	**LOAD**	E9F6
DEF	D417	**LOCATE**	C2D2
DEFINT	D618	**LOG**	D52A
DEFREAL	D61C	**LOG10**	D525
DEFSTR	D614	**LOWER$**	F834
DEG	D4E7	**MAX**	D1EE
DELETE	E728	**MEMORY**	F4EF
DI	C8E1	**MERGE**	EAA6
DIM	D67D	**MID$**	F993
DRAW	C4C6	**MIN**	D1EA
DRAWR	C4CB	**MODE**	C24F
EDIT	CØ52	**MOVE**	C505
EI	C8E7	**MOVER**	C50A
ELSE	E8F3	**NEXT**	C5FB
END	CB65	**NEW**	C12B
ENT	D385	**ON**	C7E3
ENV	D34E	**ON BREAK**	C8CB
EOF	C417	**ON ERROR**	CBF8
ERASE	D9C0	**ON SQ**	C940
ERR	D0DC	**OPENIN**	D25F
ERROR	CA8F	**OPENOUT**	D256
EVERY	C979	**ORIGIN**	C48C
EXP	D520	**OUT**	F177
FIX	FDE8	**PAPER**	C20A

DIRECCIONES DE EJECUCIÓN DE LAS PALABRAS-CLAVE DEL BASIC

Palabra-clave	*Dirección*	*Palabra-clave*	*Dirección*
PEEK	F158	SPEED	D494
PEN	C212	SQ	D329
PI	D4DB	SQR	D4EF
PLOT	C4D0	STOP	CB5A
PLOTR	C4D5	STR$	F91E
POKE	F15F	STRING$	FA36
POS	C276	SYMBOL	F69D
PRINT	F1FD	TAG	C319
'(REM)	E8F3	TAGOFF	C320
RAD	D4EB	TAN	D539
RANDOMIZE	D559	TEST	C4E9
READ	DCEB	TESTR	C4EE
RELEASE	D31E	TIME	D0E5
REM	E8F3	TROFF	DDE6
REMAIN	C99F	TRON	DDE2
RENUM	E7DF	UNT	FEC2
RESTORE	DCD9	UPPER$	F842
RESUME	CC03	VAL	FA77
RETURN	C70F	VPOS	C262
RIGHT$	F943	WAIT	F17D
RND	D584	WEND	C776
ROUND	D219	WHILE	C747
RUN	E9BD	WIDTH	C3E3
SAVE	EC09	WINDOW	C2E1
SGN	FF02	WRITE	F47B
SIN	D52F	XPOS	D107
SOUND	D2C0	YPOS	D10E
SPACE$	FA57	ZONE	F1F6

Expansión ROM

byte 0	TIPO ROM
byte 1	MARCA
byte 2	VERSIÓN
byte 3	NIVEL
byte 4	TABLA

Streams

byte 0	VIDEO
byte 1	CURSOR
byte 2	POSICIÓN CURSOR
byte 3	TAMAÑO VENTANA
byte 4	TINTA
byte 5	CARÁCTER
byte 6	GRÁFICO

Cola sonora

byte 0	CANALES QUE USAN UN ENCUENTRO
byte 1	ENVOLVENTE DE AMPLITUD
byte 2	ENVOLVENTE DE TIMBRE
bytes 3 y 4	PERIODO SONIDO
byte 5	PERIODO RUIDO
byte 6	AMPLITUD INICIAL
bytes 7 y 8	DURACIÓN DE LA ENVOLVENTE

Bloque de control de la amplitud o del timbre

byte 0	NÚMERO DE SECCIONES
bytes 1, 2 y 3	PRIMERA SECCIÓN
bytes 4, 5 y 6	SEGUNDA SECCIÓN
bytes 7, 8 y 9	TERCERA SECCIÓN
bytes 10, 11 y 12	CUARTA SECCIÓN
bytes 13, 14 y 15	QUINTA SECCIÓN

LOS BLOQUES DE CONTROL

Vector tinta

byte 0	COLOR DEL MARCO
byte 1	COLOR DE LA TINTA 0
byte 2	COLOR DE LA TINTA 1
.. ..	.. ..
byte 16	COLOR DE LA TINTA 15

Formato de los dos bytes que siguen a un RESTART

BIT 15	14	13......0
X	Y	DIRECCIÓN

ROM estándar

X=0 X=1	ROM SUPERIOR DESCONECTADA ROM SUPERIOR SELECCIONADA
Y=1 Y=0	ROM INFERIOR DESCONECTADA ROM INFERIOR SELECCIONADA

ROM suplementaria

XY da un valor de 0 a 3 que, unido a la dirección de selección de la ROM principal, proporciona la dirección de la ROM secundaria.

Formato de los ficheros cassette

El bloque completo

MOTOR GAP *	grabación ENCABEZAMIENTO DE BLOQUE	grabación DATOS

El primer bloque y el último tienen además un GAP * que permite la separación de dos programas o ficheros.

Primer bloque	MOTOR GAP	GAP DE COMIENZO	ENCABEZAMIENTO	DATOS
Último bloque	MOTOR GAP	ENCABEZAMIENTO	DATOS	GAP DE FIN

* GAP: periodo de deslizamiento sin escritura.

Formato de un registro

ARRANQUE	SEGMENTO 1	SEGMENTO 2	.....	SEGMENTO n	CIERRE

1 SEGMENTO: 256 bytes + 2 bytes de control (CRC).

Registro de cabecera : 1 SEGMENTO.
Registro de un dato : 1 a 8 SEGMENTOS (generalmente 8).

Arranque : 2048 bits a 1 seguido de un bit a 0 y de un byte de sincronismo.

Cierre : 32 bits a 1.

Formato de la cabecera

bytes 0 a 15	NOMBRE DEL FICHERO
byte 16	NÚMERO DEL BLOQUE
byte 17	# 0 SI ES EL ÚLTIMO BLOQUE
byte 18	TIPO DE FICHERO
bytes 19 y 20	LONGITUD DEL REGISTRO DATA
bytes 21 y 22	DIRECCIÓN DE ESCRITURA DATA
byte 23	# 0 SI ES EL PRIMER BLOQUE
bytes 24 y 25	LONGITUD TOTAL DEL FICHERO EN BYTES
bytes 26 y 27	PUNTO DE ENTRADA
bytes 28 a 63	NO UTILIZADOS

Descripción del byte 18 (tipo de fichero):

bit 0	1 si el fichero está protegido
bits 1 y 2	00 = BASIC 01 = BINARIO 10 = IMAGEN EN PANTALLA (DUMP) 11 = ASCII
bit 3	No usado
bits 4 a 7	Puestos siempre a 0, salvo en el caso de ficheros ASCII para los que el bit 4 está puesto a 1

Bloque de suceso

bytes 0 y 1	PUNTERO DE SISTEMA
byte 2	CONTADOR
byte 3	CLASE
bytes 4 y 5	DIRECCIÓN DE LA RUTINA DE TRATAMIENTO
byte 6	DIRECCIÓN DE LA SELECCIÓN DE LA ROM

LOS BLOQUES DE CONTROL

Bloque de control de interrupción normal

bytes 0 y 1	PUNTERO SISTEMA
bytes 2 y 3	CONTADOR. Cuando alcanza 0, tiene lugar la interrupción
bytes 4 y 5	RECARGA. Valor de reinicialización después de haber alcanzado 0
bytes 6...	BLOQUE DE SUCESO (véase más arriba)

Bloque de interrupción rápida y de interrupción CRT

bytes 0 y 1	PUNTERO DEL SISTEMA
bytes 2...	BLOQUE DE SUCESO (véase más arriba)

CIRCUITO AY3-8912 (PSG)

Estructura interna

El PSG está compuesto de los siguientes elementos:

- *Generadores sonoros:* existen tres generadores sonoros, que producen una señal de onda cuadrada de frecuencia programable. Se les llama CANALES A, B y C. No tienen prioridad propia y son independientes.
- *Generador de ruido blanco:* produce un ruido de amplio espectro.
- *Mezclador:* permite mezclar (combinar) las salidas de los tres generadores sonoros y del generador de ruido.
- *Controlador de amplitud:* permite seleccionar la amplitud de salida de la señal de dos maneras distintas. La primera, controlando la amplitud por medio del propio microprocesador y se le llama de amplitud fija. La segunda, controlando la amplitud por medio del generador de envolvente y se le llama de amplitud variable.
- *Generador de envolvente:* produce una envolvente de modulación de amplitud. Cuenta con ocho formas de envolvente.
- *Convertidores analógico-digitales:* los tres convertidores A/D producen señales a 16 niveles tal como las determina el controlador de amplitud.
- *Puerta de entrada/salida:* no sirve para la producción de sonido, y se analizará al final de este capítulo.

Los diferentes registros del PSG

Hay 15 registros, numerados desde R0 a R14. El registro R14 sirve para gestionar la puerta de entrada/salida y se analiza a continuación.

Para producir un sonido, se debe cargar con los datos una combinación de los registros R0 a R13. Cada parámetro debe ser analizado para disociar la componente de ruido, la componente de sonido, la frecuencia, la forma y la duración de la envolvente.

Una vez realizado este análisis, ya se pueden cargar los registros con los datos, y se producirá el sonido.

Los registros R0 a R5

Los tres primeros pares de registros (R0-R1, R2-R3, R4-R5) son los registros de control de la frecuencia de los tres canales A, B y C.

Los registros R0, R2 y R4 son los registros de reglaje fino y se utilizan los ocho bits. Los registros R1, R3 y R5 son los registros de reglaje grueso (sólo se utilizan los cuatro bits de izquierda LSB).

Así, los valores cargados en R0, R2 y R4 están comprendidos entre 0 y 255; los valores cargados en R1, R3 y R5 están comprendidos entre 0 y 15.

El valor se determina por la fórmula siguiente:

VL = 125000 / F

El registro R6

El registro R6 determina la frecuencia del generador de ruido; sólo se utilizan los cinco bits menos significativos. Así pues, el valor de R6 está comprendido entre 1 y 31. Se utiliza la misma fórmula que para R0-R5.

El registro R7

El registro R7 controla la mezcla entre los tres generadores sonoros y el generador de ruido. R7 sirve también para controlar la puerta de la que se habla más adelante.

He aquí una tabla resumen de los efectos del registro R7.

BIT	=0	=1
7	NO SE USA	NO SE USA
6	PUERTA DE ENTRADA	PUERTA DE SALIDA (no se usa)
5	RUIDO EN EL CANAL C ON	RUIDO EN EL CANAL C OFF
4	RUIDO EN EL CANAL B ON	RUIDO EN EL CANAL B OFF
3	RUIDO EN EL CANAL A ON	RUIDO EN EL CANAL A OFF
2	SONIDO EN EL CANAL C ON	SONIDO EN EL CANAL C OFF
1	SONIDO EN EL CANAL B ON	SONIDO EN EL CANAL B OFF
0	SONIDO EN EL CANAL A ON	SONIDO EN EL CANAL A OFF

Nota: no hay bastante con poner un canal en OFF para detener su emisión; es necesario, además, escribir 0 en el registro de control de amplitud.

Ejemplo

Se desea en el canal A: sólo sonido sin ruido; en el B: sonido y ruido; en el C: sólo ruido.

valor : x x 0 0 1 1 0 0 = 12
bit : 7 6 5 4 3 2 1 0
(x x) = sin importancia

Bastará con escribir 12 en el registro 7.

Los registros R8 a R10

Los registros R8 a R10 controlan las amplitudes de los canales A, B y C; sólo se usan los cuatro bits menos significativos, pues los posibles valores están comprendidos entre 0 y 15. 0 significa que la amplitud es mínima (nula) y 15 corresponde a la amplitud máxima. El quinto bit (bit 4) es el bit de selección del modo de funcionamiento del control de amplitud. Si el BIT 4 está a 0, la amplitud no varía. Si el BIT 4 está a 1, la amplitud viene controlada por el generador de envolvente *(ver abajo).*

Los registros R11 y R12

Estos dos registros controlan el periodo de la envolvente. Se efectúa un cálculo con una fórmula parecida a la utilizada para R0-R5 para determinar el valor de R11 y R12.

Fórmula: VL = 125000 * P / 16, donde P es el periodo de la envolvente.

El registro R13

El registro R13 controla la forma de modulación utilizada. Si el BIT 4 descrito en los registros R8 a R10 está a 1, tiene lugar la modulación, de lo contrario, se ignora la programación del registro R13.

Solamente se usan los cuatro bits menos significativos.

Bit 3	2	1	0	*Forma de la envolvente*	*Valores posibles*
0	0	x	x	A Empieza un solo ciclo con la amplitud máxima que disminuye hasta llegar a ser nula.	0, 1, 2, 3
0	1	x	x	B Empieza un solo ciclo con la amplitud nula que aumenta hasta alcanzar el valor máximo, y luego vuelve bruscamente a 0.	4, 5, 6, 7
1	0	0	0	C Como en A, pero se va repitiendo sin parar.	8

CIRCUITOS

Bit 3	2	1	0	*Forma de la envolvente*	*Valores posibles*
1	0	1	0	D Como C, pero sube de modo más marcado hacia el máximo (ATTACK).	10
1	0	1	1	E Como A, pero llega en seguida al máximo, y permanece en él.	11
1	1	0	0	F Como B, pero se repite sin parar.	12
1	1	0	1	G Como B, pero permanece en el máximo.	13
1	1	1	0	H Como F, pero con un ataque más marcado.	14

El registro R14

Este registro no tiene nada que ver con la producción de sonido. Gestiona la puerta de entrada/salida que se encarga de la lectura del teclado y de la manecilla de juegos.

El bit 6 del registro R7 regula el sentido de la transmisión, pero como la puerta se utiliza solamente como entrada, basta con poner a 0 el bit 6 de R7.

Programación del AY3-8912

El PSG es asequible a través de las puertas A y C del PPI 8255 *(véase "Circuito PPI 8255", página 143).*

La escritura en el PSG puede hacerse fácilmente por medio de la rutina 188 (BD34). Al ser, pues, más difícil realizar directamente la lectura del estado del teclado y de las manecillas de juego, se aconseja hacerlo por los puntos de entrada estándar del software del sistema.

Para los que quieran a pesar de todo programar directamente el PSG les señalamos que las dos señales de la orden BDIR y BC1 las proporciona la puerta C del PPI 8255.

Función de BDIR y BC1

BDIR	*BC1*	*Función*
0	0	Inactivo: no hay función.
0	1	Lectura: el contenido del registro actual es enviado al bus de datos D0-D7.
1	0	Escritura: el bus D0-D7 contiene el dato a escribir en el registro actual.
1	1	Escritura: el bus D0-D7 contiene el número de registro que vendrá a ser el registro actual.

Generalidades

El **PPI** es un circuito fabricado por INTEL con la denominación **8255A**. Es un circuito de interface previsto para los procesadores de la familia 8080.

Posee 24 bits de entrada/salida que pueden programarse en dos grupos de 12 bits, y utilizarse de tres modos principales.

En el primer modo (modo 0), cada grupo de 12 bits puede estar programado por secciones de 4 bits tanto en entrada como en salida.

En el segundo modo (modo 1), cada grupo de 12 bits puede programarse de la siguiente manera: 8 bits se usan en entrada/salida y los otros 4 para el HANDSHAKING (control de transmisión).

En el tercer modo (modo 2) se utilizan 8 bits como PUERTA bidireccional y 5 bits para el HANDSHAKING.

El PPI cuenta también con la posibilidad de posicionar directamente los bits al estado 1 o al 0.

Para más facilidad, el PPI está dividido en tres puertas de 8 bits distintos, denominadas PUERTA A, PUERTA B y PUERTA C.

La PUERTA C se divide en dos grupos de 4 bits para formar los grupos de 12 bits con A y B.

Desglose de las PUERTAS

Puerta A — usada en entrada y en salida

B0 a B7	Correspondiente a los D0 a D7 del AY3-8912.

Puerta B — usada en entrada solamente

Bit 7	Lectura de datos del cassette (INPUT).
Bit 6	Señal de ocupado (BUSY) de la impresora (INPUT).
Bit 5 Bit 4 Bit 3 Bit 2 Bit 1	No utilizables.
Bit 0	Interrupción proveniente del CRT.

C
I
R
C
U
I
T
O
S

Puerta C — usada en salida solamente

Bit 7	Orden BDIR del AY3-8912 OUT.
Bit 6	Orden BC1 del AY3-8912 OUT.
Bit 5	Escritura de datos en el cassette.
Bit 4	Arranque del motor del cassette.
Bits 3 a 0	Selección de la línea del teclado.

Programación

El PPI actúa de interface con las direcciones siguientes:

Dirección F4xx	Lectura y escritura PUERTA A.
Dirección F5xx	Lectura y escritura PUERTA B.
Dirección F6xx	Lectura y escritura PUERTA C.
Dirección F7xx	Escritura en el registro de control.

Notas:
- xx significa cualquier valor.
- A se usa en escritura y lectura, B se usa solamente en lectura y C sólo en escritura.

De los tres modos descritos brevemente en las Generalidades, sólo se estudiará el MODO 0, puesto que basta para todas las manipulaciones previstas.

El PPI es programable a través de un registro de control en el que solamente se puede escribir. No está permitida ninguna lectura de este registro.

Escritura en el registro de control

En el registro de control se escribe por medio de un simple OUT en la PUERTA F7xx.

La palabra de control es una palabra de 8 bits de la que mostramos su funcionamiento bit por bit.

Bit 7	Siempre 1 si se trata de una palabra de control.
Bit 6	Determinación del modo de funcionamiento del grupo A. Para seleccionar el MODO 0, este bit debe estar a 0. Si está a 1 se selecciona el MODO 2.
Bit 5	Determinación del modo de funcionamiento del grupo A. Para seleccionar el MODO 0, este bit debe estar a 0. Si está a 1 se selecciona el MODO 1.
Bit 4	Determinación del sentido de funcionamiento de la PUERTA A. 0 significa EN SALIDA y 1 significa EN ENTRADA. Será siempre 1.

Bit 3	Determinación del sentido de funcionamiento de la parte alta de la PUERTA C. 0 significa EN SALIDA y 1 significa EN ENTRADA.
Bit 2	Determinación del modo de funcionamiento del grupo B. 0 significa MODO 0 y 1 significa MODO 1. Será siempre 0.
Bit 1	Determinación del sentido de funcionamiento de la PUERTA B. 0 significa EN SALIDA y 1 significa EN ENTRADA. Será siempre 1.
Bit 0	Determinación del sentido de funcionamiento de la parte baja de la PUERTA C. 0 significa EN SALIDA y 1 significa EN ENTRADA. Será siempre 0.

Si el bit 7 es igual a 0, el registro sólo es utilizado como controlador de las PUERTAS, pero permite posicionar los bits de la PUERTA C a 1 o a 0.

Bit 7 = 0 : funcionamiento en posicionamiento de bit.
Bits 6, 5 y 4 : no se usan.
Bits 3, 2 y 1 : dan el número del bit a posicionar.
Bit 0 : da el sentido del posicionamiento, 1 significa posicionamiento del bit a 1 y 0 significa del bit a 0.

La programación se hace, pues, enviando la palabra de estado que convenga al registro de control y efectuando una lectura o una escritura en la PUERTA idónea.

CIRCUITO CRT 6845

Generalidades

El circuito **6845** controla la generación de las señales de video. Posee una PUERTA bidireccional de 8 bits y puede posicionarse por medio de 19 registros internos. Uno de los registros sirve además de buffer para la programación de los otros 18.

Los distintos registros del 6845

R0 a R3	Determinan el formato horizontal y el timing. En estándar se cargan con los valores requeridos en función del modo. Por ejemplo, en modo 1: R0=63, R1=40, R2=46, R3=142.
R4 a R9	Determinan el formato vertical. Se cargan con los valores requeridos. R4=38, R5=0, R6=25, R7=30.
R10 a R15	Gestionan el cursor y están continuamente modificados por el software.
R16 a R17	Se ocupan de la gestión del lápiz óptico (no implementado).

R0	Número de caracteres total en horizontal 0-255.	
R1	Número de caracteres visualizados en horizontal 0-255.	
R2	Sincronismo horizontal (posición) 0-255.	
R3	Longitud de sincronismo 0-15.	
R4	Número de líneas total en vertical 0-127.	
R5	Sincronismo vertical 0-31.	
R6	Número de caracteres visualizados en vertical 0-127.	
R7	Sincronismo vertical (posición) 0-127.	
R8	Modo entrelazado 0-3.	
R9	Scanning 0-31.	
R10	Línea de partida del scanning del cursor 0-31.	
R11	Línea de final del scanning del cursor 0-31.	
R12	Byte más significativo	de la dirección de partida.
R13	Byte menos significativo	de la video-ram a 16383.
R14	Posición del cursor (OLPS)	0-16383.
R15	Posición del cursor (OLMS).	

Programación

Bastan dos direcciones de PUERTA para programar el CRT.

La PUERTA BCxx sirve para dar las direcciones de registro y la PUERTA BDxx sirve para escribir los datos en el registro actual.

Los registros son sólo de escritura, con excepción del R14 y del R15 que pueden ser leídos, y así dan la posición actual del cursor.

Generalidades

El **Amstrad** está equipado con un circuito especial que se encarga de la conmutación de las ROM y del control del CRT 6845. Este circuito no lleva un número estándar y se le llama **GATE ARRAY**.

Programación

La GATE ARRAY puede considerarse como una puerta de salida de 8 bits gobernada por un OUT en la PUERTA 7Fxx.

Los dos bits superiores controlan el tipo de aplicación.

B7	*B6*	
0	0	Carga del registro de paleta.
0	1	Carga de la memoria de paleta.
1	0	Conmutación de ROM y control de video.
1	1	Reservado.

Conmutación de ROM y control de video

BIT 7=1, BIT 6=0, BIT 5=0.

BIT 4=1 : pondrá a 0 el divisor que genera las interrupciones.

BIT 3=0 : ROM superior conectada; BIT 3=1 : ROM superior desconectada.

BIT 2=0 : ROM inferior conectada; BIT 2=1 : ROM inferior desconectada.

BIT 1 : control de video MC1.

BIT 0 : control de video MC0.

MC1	*MC0*	
0	0	Modo 0 (20 × 24).
0	1	Modo 1 (40 × 24).
1	0	Modo 2 (80 × 24).
1	1	No utilizable.

Registro de paleta

BIT 7=0, BIT 6=0, BIT 5=0.

BIT 4=0 : carga del número de color de la tinta dado por B0-B3.

BIT 4=1 : carga del número de color del borde (B0-B3 ignorados).

BIT 3 a BIT 0 dan el número de la tinta (15 posibles colores).

Memoria de paleta

BIT 7=0, BIT 6=1, BIT 5=0.

BIT 4 a BIT 0 : 31 valores para la decodificación del color de registro de paleta. El número de colores posibles varía en función del modo elegido.

DUMP HEXA DE LA MEMORIA ROM INFERIOR Y SUPERIOR EN LA IMPRESORA

Estos programas permiten obtener en la impresora el contenido de las memorias ROM en hexadecimal.

◇ Dump hexa memoria ROM inferior

```
10 MEMORY &6000
15 CLS
20 FOR I=&A000 TO &A010
30 READ A$
40 POKE I,VAL("&H"+A$)
50 NEXT I
60 DATA F3,CD,06,B9,21,00,00,11,00,60,01,FF,3F,ED,B0,C9
70 DATA 00
80 CALL &A000
100 FOR I=&6000 TO 40960
120 IF INT(I/16)*16=I THEN PRINT #8,"":
    PRINT #8,HEX$(I-&6000);"  ";
130 A=PEEK(I)
135 A$=RIGHT$("00"+HEX$(A),2)
140 PRINT #8,A$;"   ";
150 NEXT I
```

◇ Dump hexa memoria ROM superior

```
10 MEMORY &6000
15 CLS
20 FOR I=&A000 TO &A010
30 READ A$
40 POKE I,VAL("&H"+A$)
50 NEXT I
60 DATA F3,CD,00,B9,21,00,C0,11,00,60,01,FF,3F,ED,B0,C9
70 DATA 00
80 CALL &A000
100 FOR I=&6000 TO 40960
120 IF INT(I/16)*16=I THEN PRINT #8,"":
    PRINT #8,HEX$(I+&6000);"  ";
130 A=PEEK(I)
135 A$=RIGHT$("00"+HEX$(A),2)
140 PRINT #8,A$;"  ";
150 NEXT I
```

DUMP ASCII DE LA MEMORIA ROM INFERIOR Y SUPERIOR EN LA IMPRESORA

◇ Dump ASCII memoria ROM inferior

```
10 MEMORY &6000
15 CLS
20 FOR I=&A000 TO &A010
30 READ A$
40 POKE I,VAL("&H"+A$)
50 NEXT I
60 DATA F3,CD,06,B9,21,00,00,11,00,60,01,FF,3F,ED,B0,C9
70 DATA 00
80 CALL &A000
100 FOR I=&6000 TO 40960
120 IF INT(I/64)*64=I THEN PRINT #8,"":
    PRINT #8,HEX$(I-&6000);"  ";
130 A=PEEK(I)
140 IF (A>31 AND A<127) OR A>159 THEN PRINT #8, CHR$(A);
    ELSE PRINT #8,".";
150 NEXT I
```

◇ Dump ASCII memoria ROM superior

```
10 MEMORY &6000
15 CLS
20 FOR I=&A000 TO &A010
30 READ A$
40 POKE I,VAL("&H"+A$)
50 NEXT I
60 DATA F3,CD,00,B9,21,00,C0,11,00,60,01,FF,3F,ED,B0,C9
70 DATA 00
80 CALL &A000
100 FOR I=&6000 TO 40960
120 IF INT(I/64)*64=I THEN PRINT #8,"":
    PRINT #8,HEX$(I+&6000);"  ";
130 A=PEEK(I)
140 IF (A>31 AND A<127) OR A>159 THEN PRINT #8, CHR$(A);
    ELSE PRINT #8,".";
150 NEXT I
```

ARRANQUE Y PARO DEL MOTOR DEL CASSETTE

Arranque : OUT &HF600,16
Paro : OUT &HF600,0

PROTECCIÓN DEL PROGRAMA

Teclear como primera línea : 10 REM
como segunda línea : 20 PRINT "COMIENZO"

a continuación, escribir el programa que deba protegerse.

Cuando el programa está enteramente codificado, teclear:

POKE 372,225

A partir de este momento no es posible listar el programa y solamente un RUN 20 permite lanzar la ejecución.

El POKE 372,225 tiene como efecto sustituir en la memoria el token de la instrucción REM por un token inexistente (255).

De esta manera, cuando el ordenador intenta listar el programa, encuentra un token que no puede interpretar y visualiza un SYNTAX ERROR.

Igualmente, cuando intenta ejecutar el programa (RUN), se encuentra en la primera línea con un token inexistente y se detiene. Solamente un RUN 20 permite lanzar la ejecución del programa, puesto que evita la lectura de la línea 10 que contiene el token inexistente.

RUIDOS ESPECIALES

```
5 REM SIRENA DE POLICIA
10 FOR I=80 TO 220 STEP 12
20 SOUND 1,I,2
30 NEXT I
40 FOR I=220 TO 80 STEP -12
50 SOUND 1,I,2
60 NEXT I
70 GOTO 10
```

```
5 REM TIRO DE FUSIL
10 FOR I=90 TO 125
20 SOUND 1,I,2,15
30 NEXT I
50 GOTO 10
```

RUIDOS ESPECIALES

```
5 REM TOQUE DE DIFUNTOS
10 FOR I=15 TO 8 STEP -1
20 SOUND 1,500,20,I,,,1
30 NEXT I
```

PROGRAMA QUE PERMITE TRAZAR CÍRCULOS Y ELIPSES

Este programa permite simular la instrucción CIRCLE del Basic Microsoft con la que no cuenta el Basic Amstrad.

X e **Y** representan las coordenadas horizontal y vertical del centro del círculo.

R representa el radio que hay que dar al círculo.

AD representa el ángulo de partida y **AF** el ángulo de llegada. Los dos se expresan en grados y permiten dibujar arcos de círculo.

FAP representa el factor de aplanamiento, y permite dibujar elipses.

```
5 REM DIBUJA ELIPSES
10  CLS
20 X=320:Y=200:R=100
30 AD=0
40 AF=360
50 FAP=2
60 DEG
70 PLOT X+R*COS(AD),Y+R*SIN(AD)
80 FOR A=AD TO AF
90 X1=X+R*COS(A):Y1=Y+R*SIN(A)/FAP
100 DRAW X1,Y1
110 PLOT X1,Y1
120 NEXT A
```

SCANNING DEL TECLADO

Escribir el programa que sigue, ejecutarlo y pulsar diferentes teclas. Observar los valores así obtenidos, que se podrán utilizar en los propios programas realizando un PEEK del byte deseado y comprobando su valor. Esta rutina sustituye con ventaja la instrucción INKEY$.

```
10 FOR I=&B4EB TO &B4F4
20 PRINT PEEK(I);
30 NEXT I
40 PRINT
50 GOTO 10
```

MODIFICACIÓN ESPECIAL DEL COLOR DE FONDO

El POKE siguiente modifica el color de fondo (PAPER). Probar distintos valores de N.

POKE &B290,N

Nota: N debe estar comprendido entre 0 y 255.

INSTALACIÓN DE UNA RUTINA EN LENGUAJE MÁQUINA EN UNA INSTRUCCIÓN REM

Las rutinas muy cortas y que no contengan 2 bytes consecutivos puestos a 0 pueden ser implantadas en una línea REM.

Escribir:

10 REM **

Poner tantos bytes como asteriscos en la rutina.

El Basic empieza en 368 y el primer asterisco en 374.

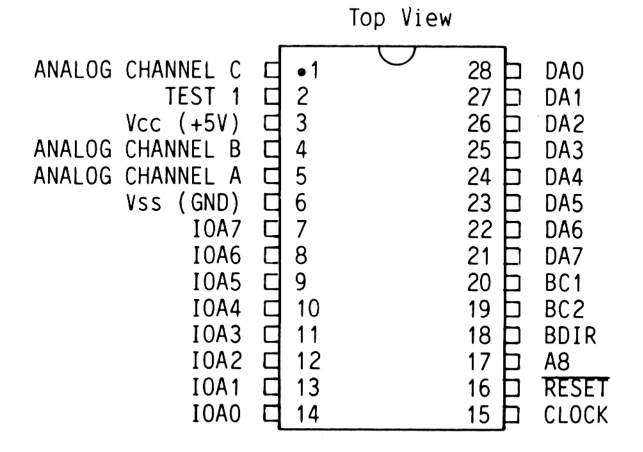
El programa siguiente instala la rutina y se borra a continuación.

```
20 FOR I=374 TO 379: REM SI LA RUTINA TIENE 6 BYTES DE LARGO
30 READ A$
40 POKE I,VAL("&H"+A$)
50 NEXT I
60 DATA 3E,19,21,88,CD,C9
70 DELETE 20-70
```

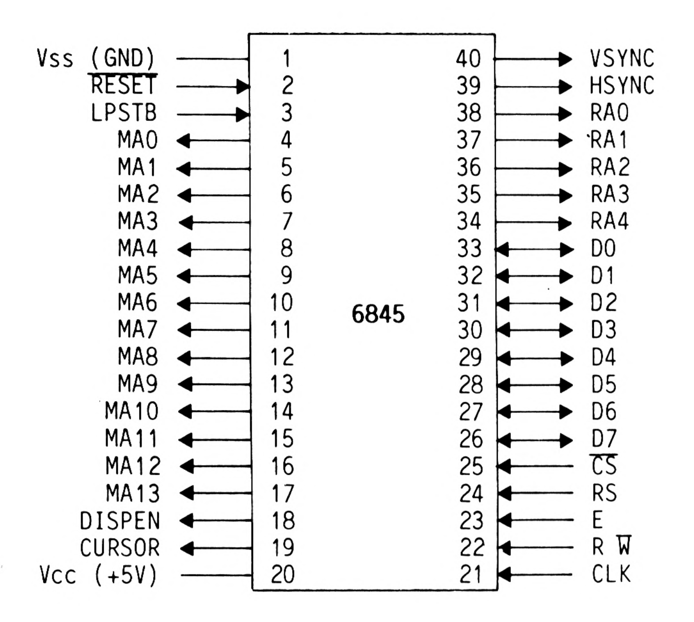
Nota: la rutina instalada en el programa anterior es a título de ejemplo, porque en realidad no hace nada de particular.

CONEXIONES DEL AY3-8912

28 LEAD DUAL IN LINE
AY3-8912



CONEXIONES DEL CRT 6845



Nombre de la patilla	*Descripción*	*Tipo*
D0-D7	Bus de datos.	Bidireccional Triestado
CS	Selección del circuito.	Input
RS	Selección del registro.	Input
R/W	Lectura/escritura.	Input
E	Señal de sincronismo.	Input
CLK	Reloj.	Input
RESET	Inicialización.	Input
Vcc	Alimentación (+5V).	Input
MA0-MA13	Dirección de memoria (16K).	Output
RA0-RA4	Dirección de línea (scanning).	Output
HSYNC	Sincronismo horizontal.	Output
VSYNC	Sincronismo vertical.	Output
DISPEN	Validación de visualización.	Output
CURSOR	Validación del cursor.	Output
LPSTB	Presencia lápiz óptico.	Input

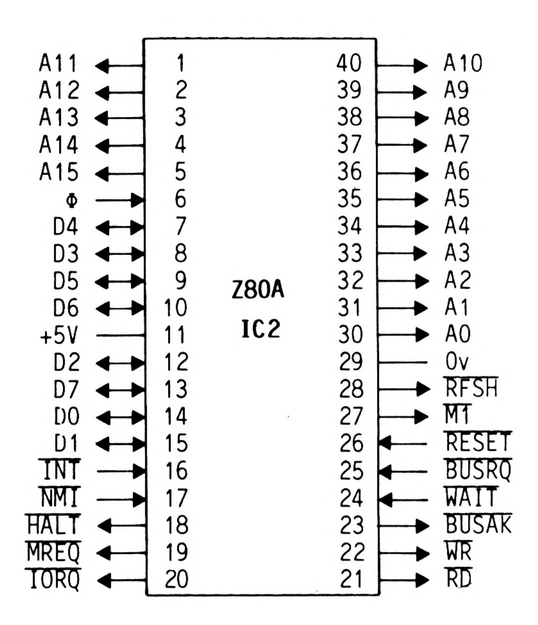
CONEXIONES DEL PPI 8255

8255A

PA3	1	40	PA4
PA2	2	39	PA5
PA1	3	38	PA6
PA0	4	37	PA7
$\overline{RD}$	5	36	$\overline{WR}$
$\overline{CS}$	6	35	RESET
GND	7	34	D0
A1	8	33	D1
A0	9	32	D2
PC7	10	31	D3
PC6	11	30	D4
PC5	12	29	D5
PC4	13	28	D6
PC0	14	27	D7
PC1	15	26	Vcc
PC2	16	25	PB7
PC3	17	24	PB6
PB0	18	23	PB5
PB1	19	22	PB4
PB2	20	21	PB3

Nombre de la patilla	*Descripción*
D7-D0	Bus de datos (bidireccional).
RESET	Inicialización.
CS	Selección del circuito.
RD	Lectura input.
WR	Escritura input.
A0, A1	Dirección de puerta.
PA7-PA0	Puerta A (bit).
PB7-PB0	Puerta B (bit).
PC7-PC0	Puerta C (bit).
Vcc	Alimentación (+5 voltios).
GND	0 voltios.

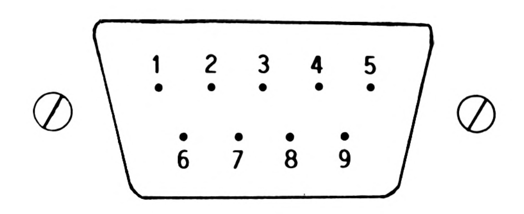
CONEXIONES DEL Z80



Patilla	*Significado*
1	Bit de dirección 11.
2	Bit de dirección 12.
3	Bit de dirección 13.
4	Bit de dirección 14.
5	Bit de dirección 15.
6	Reloj.
7	Bit de datos 4.
8	Bit de datos 3.
9	Bit de datos 5.
10	Bit de datos 6.
11	Tensión regulada de 5 voltios.
12	Bit de datos 2.
13	Bit de datos 7.
14	Bit de datos 0.
15	Bit de datos 1.
16	Interrupciones enmascarables.
17	Interrupciones no enmascarables.
18	Señal de detención del microprocesador.
19	Petición de operación memoria.
20	Petición de entradas/salidas.
21	Orden de lectura memoria.

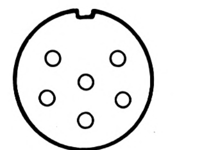
Patilla	*Significado*
22	Orden de escritura memoria.
23	Aceptación de acceso directo memoria.
24	Petición de espera al microprocesador.
25	Petición de acceso directo memoria.
26	Inicialización del microprocesador.
27	Señal de comienzo del ciclo.
28	Refresco de las memorias dinámicas.
29	0 voltios de tensión para las masas eléctricas.
30	Bit de dirección 0.
31	Bit de dirección 1.
32	Bit de dirección 2.
33	Bit de dirección 3.
34	Bit de dirección 4.
35	Bit de dirección 5.
36	Bit de dirección 6.
37	Bit de dirección 7.
38	Bit de dirección 8.
39	Bit de dirección 9.
40	Bit de dirección 10.

EL CONECTOR PARA LA MANECILLA DE JUEGOS



Borne 1	arriba	Borne 6	botón de tiro 2
Borne 2	abajo	Borne 7	botón de tiro 1
Borne 3	izquierda	Borne 8	masa común
Borne 4	derecha	Borne 9	masa común 2
Borne 5	disponible		

EL CONECTOR DE SALIDA DE VIDEO



5 1
6
4 2
3

Borne 1	rojo	Borne 4	sincronismo
Borne 2	verde	Borne 5	masa
Borne 3	azul	Borne 6	luminancia

EL CONECTOR DE SALIDA EXPANSIÓN

49 47 45 43 41 39 37 35 33 31 29 27 25 23 21 19 17 15 13 11 9 7 5 3 1

50 48 46 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2

Borne 1	sonido	Borne 26	D0
Borne 2	masa	Borne 27	+ 5 voltios
Borne 3	A15	Borne 28	$\overline{\text{MREQ}}$
Borne 4	A14	Borne 29	$\overline{\text{M1}}$
Borne 5	A13	Borne 30	$\overline{\text{RFSH}}$
Borne 6	A12	Borne 31	$\overline{\text{IORQ}}$
Borne 7	A11	Borne 32	$\overline{\text{RD}}$
Borne 8	A10	Borne 33	$\overline{\text{WR}}$
Borne 9	A9	Borne 34	$\overline{\text{HALT}}$
Borne 10	A8	Borne 35	$\overline{\text{INT}}$
Borne 11	A7	Borne 36	$\overline{\text{NMI}}$
Borne 12	A6	Borne 37	$\overline{\text{BUSRD}}$
Borne 13	A5	Borne 38	$\overline{\text{BUSAK}}$
Borne 14	A4	Borne 39	READY
Borne 15	A3	Borne 40	$\overline{\text{BUS RESET}}$
Borne 16	A2	Borne 41	$\overline{\text{RESET}}$
Borne 17	A1	Borne 42	$\overline{\text{ROMEN}}$
Borne 18	A0	Borne 43	ROMDIS
Borne 19	D7	Borne 44	$\overline{\text{RAMRD}}$
Borne 20	D6	Borne 45	RAMDIS
Borne 21	D5	Borne 46	CURSOR
Borne 22	D4	Borne 47	L. PEN
Borne 23	D3	Borne 48	$\overline{\text{EXP}}$
Borne 24	D2	Borne 49	masa
Borne 25	D1	Borne 50	0

EL CONECTOR DE SALIDA PARA IMPRESORA

Borne 1	STROBE	Borne 19	GND
Borne 2	D0	Borne 20	GND
Borne 3	D1	Borne 21	GND
Borne 4	D2	Borne 22	GND
Borne 5	D3	Borne 23	GND
Borne 6	D4	Borne 24	GND
Borne 7	D5	Borne 25	GND
Borne 8	D6	Borne 26	GND
Borne 9	D7	Borne 28	GND
Borne 11	BUSY	Borne 33	GND
Borne 14	GND (masa)	Los bornes no descritos	
Borne 16	GND	no están conectados.	

ANEXOS

INSTRUCCIONES Y FUNCIONES PROPIAS DEL CPC664

Funciones

COPYCHR$

COPYCHR$ (# número de canal)
Copia en una variable alfanumérica el carácter actual bajo el cursor del canal especificado.

DEC$

DEC$ (expresión numérica, formato)
Escribe la expresión numérica en el formato especificado (este formato es idéntico al de la instrucción PRINT USING). Esta función permite poner el resultado de un USING en una variable alfanumérica.

DERR

DERR
Imprime el último número de error producido.

SPC

SPC(n)
Genera espacios. Se usa con PRINT.

Instrucciones

CLEAR INPUT

CLEAR INPUT
Esta instrucción vacía el buffer de entrada, eliminando todos los caracteres parásitos que pueda hallar en él.

CURSOR

CURSOR bit sistema, bit usuario
Esta instrucción permite provocar la extinción o el encendido del cursor. Si el bit vale 1 el cursor está encendido, si vale 0, no lo está.

FILL

FILL tinta
Rellena la superficie limitada con el color de la tinta especificado.

FRAME

FRAME
Sincroniza la escritura de los gráficos con el retorno del barrido.

GRAPHICS

GRAPHICS PAPER tinta / GRAPHICS PEN tinta
Posiciona la tinta del papel gráfico o de la pluma gráfica sin alterar la pluma o el papel de texto.

MASK

MASK número entero de 0 a 255, número de 0 a 7
Una bonita instrucción que permite especificar la estructura de un trazo para dibujar en punteado o en trazo mixto. El primer byte especifica la estructura del trazo en 8 puntos (0

a 255), el segundo especifica el punto de partida en el interior de los 8 puntos.
Ejemplo: para dibujar en punteado un punto cada dos:
MASK &X10101010,0 o MASK 170,0

MID$ a izquierda

MID$ (cadena1, posición, longitud) = cadena2
Sustituye en el interior de la cadena1, a partir del carácter definido por posición y con la longitud definida por longitud, los caracteres actuales por los de la cadena1.

ON BREAK CONT

ON BREAK CONT
Impide la interrupción del programa al pulsar la tecla BREAK. Esta función hay que usarla con precaución en los programas terminados. Una vez puesto en marcha el único modo de detener el programa es por medio de RESET.

LOS VECTORES DE LLAMADA A LAS RUTINAS MATEMÁTICAS DEL CPC664

Las rutinas matemáticas contenidas en la ROM inferior deben ser frecuentemente llamadas desde la ROM BASIC para realizar todas las funciones BASIC de cálculo (+, –, *, /, sen, cos...). Se han creado una serie de vectores para facilitar esta llamada.

Las funciones matemáticas del BASIC funcionan sobre un acumulador virtual de seis bytes situado en B09F a B0A4. B09F contiene el tipo de la variable (2=entero, 3=cadena, 5=número real).

Una variable entera se codifica en dos bytes en binario con signo.

Una variable real es más compleja. Se representa por medio de cinco bytes siguiendo un código binario particular:

- expresar el número en binario;
- contar el número de cifras significativas situadas delante de la coma y añadirle 128 (80H). Se obtiene así el byte 5;
- suprimir el primer bit de la izquierda y convertir los siete restantes bits a decimal. Si el número es negativo, añadir 128 (80H). Se obtiene el byte 4;
- para obtener los bytes 3, 2 y 1 tomar los bits siguientes en grupos de 8 y convertirlos a decimal.

Ejemplo: codificación de la variable real: –2527

2527 se escribe en binario 1 0011101 1111 (12 cifras).

Byte 5 = 8C porque	:	128 + 12 = 140 = 8C.
Byte 4 = 9D porque	:	tomar los siete bits siguientes: 0011101 = 29 = 1D. Al ser el número negativo, añadir 128: 29 = 128 = 157 = 9D.
Byte 3 = F0 porque	:	los ocho bits siguientes son 1111 (0000) = 240 = F0.

Byte 2 y byte 1 = 00 porque no hay más bits.

–2527 se codifica, pues: 00 00 F0 9D 8C.

LOS VECTORES DE LLAMADA A LAS RUTINAS MATEMÁTICAS DEL CPC664

Dirección vector	*Dirección real*	*Significado*
BD5E	2F91	Copia los cinco bytes apuntados por DE en la zona apuntada por HL y pasa a A el contenido del byte que se halla en la dirección HL-1 (tipo de variable).
BD61	2F9F	Conversión de entero a flotante en los cinco bytes apuntados por DE.
BD64	2FC8	Conversión del número binario apuntado por HL en número con formato del acumulador (5 bytes).
BD67	2FD9	Transforma el valor contenido en los cinco bytes apuntados por HL en un entero contenido en HL.
BD6A	3001	Transforma el valor contenido en los cinco bytes apuntados por HL en un entero contenido en los dos primeros bytes apuntados por HL.
BD6D	3014	Realiza la función FIX.
BD70	3055	Realiza la función INT.
BD73	305F	Función SGN (rutina usada por STR$ y PRINT).
BD76	30C6	Rutina de transformación (multiplica por 10 exp A).
BD79	34A2	Suma de dos números reales. HL apunta a una zona de cinco bytes que representa un número de formato real (llamada ACCUM1). DE apunta a otra zona de cinco bytes (llamada ACCUM2). A la salida de la rutina, HL apunta siempre a ACCUM1 y ACCUM1 contiene el valor de ACCUM1 + ACCUM2.
BD7C	3159	Función RND.
BD7F	349E	Sustracción de dos números reales. HL apunta a una zona de cinco bytes que representa un número de formato real (llamada ACCUM1). DE apunta a otra zona de cinco bytes (llamada ACCUM2). A la salida de la rutina, HL apunta siempre a ACCUM1, y ACCUM1 contiene el valor de ACCUM1 - ACCUM2.
BD82	3577	Multiplicación de dos números reales. Como antes, pero ACCUM1 contiene el valor de ACCUM1 * ACCUM2.
BD85	3604	División de dos números reales. Como antes, pero ACCUM1 contiene el valor de ACCUM1 / ACCUM2.

LOS VECTORES DE LLAMADA A LAS RUTINAS MATEMÁTICAS DEL CPC664

Dirección vector	*Dirección real*	*Significado*
BD88	3188	Proporciona el último valor de RND.
BD8B	36DF	Comparación de dos números reales: Si ACCUM1 > ACCUM2, entonces A=1 Si ACCUM1 < ACCUM2, entonces A=255 Si ACCUM1 = ACCUM2, entonces A=0
BD8E	3731	Negación de un número real. HL apunta a ACCUM1 que contiene el valor −ACCUM1.
BD91	3727	Comprueba el número real contenido en ACCUM1. HL apunta a ACCUM1. Si ACCUM1 > 0, entonces A=1 Si ACCUM1 < 0, entonces A=255 Si ACCUM1 = 0, entonces A=0
BD94	3345	Posicionamiento del modo de cálculo de ángulos en grados o en radianes. Si A=0 se está operando en RADIANES. Si A≠0 se está operando en GRADOS.
BD97	2F73	En salida, la zona apuntada por HL en entrada contiene la constante PI.
BD9A	32AC	Extracción de la raíz cuadrada de un número real. En entrada, HL apunta a una zona de cinco bytes que contiene un número. En salida, esta zona contiene la raíz cuadrada del número.
BD9D	32AF	Cálculo de la potenciación de un número real. HL apunta a ACCUM1 que contiene el número y DE apunta a ACCUM2 que contiene el exponente. En salida, ACCUM1 contiene el valor de ACCUM1 elevado a ACCUM2.
BDA0	31B6	Cálculo del logaritmo neperiano de un número real. HL apunta a ACCUM1 que contiene el número de entrada. En salida ACCUM1 contiene el valor del logaritmo neperiano del número.
BDA3	31B1	Cálculo del logaritmo en base 10 de un número real. HL apunta a ACCUM1 que contiene el número de entrada. En salida, ACCUM1 contiene el valor del logaritmo decimal del número.
BDA6	322F	Cálculo del exponencial de un número.

LOS VECTORES DE LLAMADA A LAS RUTINAS MATEMÁTICAS DEL CPC664

Dirección vector	*Dirección real*	*Significado*
		HL apunta a ACCUM1 que en salida contiene el valor del exponencial del número.
BDA9	3353	Cálculo del seno de un ángulo.
BDAC	3349	Cálculo del coseno de un ángulo.
BDAF	33C8	Cálculo de la tangente de un ángulo.
BDB2	33D8	Cálculo del arcotangente de un ángulo.
BDB5	2FD1	Rutina de evaluación.
BDB8	3136	Rutina RND (B8E4 y B8E6) en la inicialización.
BDBB	3143	Rutina utilizada para la generación de números aleatorios.

LAS PRINCIPALES VARIABLES DE SISTEMA DEL CPC664

Dirección	*Longitud*	*Significado*
AC01	1	Bit: 0 = auto, 1 = no auto.
AC02	2	Número de la línea actual (usada por auto).
AC04	2	Valor del incremento entre dos líneas (AUTO).
AC09	1	Usado por la instrucción WIDTH.
AC0C	1	Usado por la instrucción NEXT.
AC12	2	Usado por la instrucción FOR.
AC14	2	Usado por la pareja de instrucciones WHILE-WEND.
AC16	11	Usado por la instrucción ON... GOTO.
AC8A	256	Buffer de entrada teclado.
AD8C	2	Puntero para la instrucción RESUME.
AD8E	2	Usado en el tratamiento de error.
AD90	1	Número de error.
AD91	2	Dirección del último byte ejecutado.
AD93	2	Dirección para END, STOP y CONT.
AD98	1	Número de error para la función ON ERROR GOTO.
AD99	9	Parámetros usados por la instrucción SOUND.
ADF3	26	Tabla de declaración de las variables. Compuesta de 26 caracteres (uno por cada letra del alfabeto). Cada byte contiene un código que determina el tipo por omisión de cada variable que empiece por esta letra.
AE15	2	Dirección de la línea actual para READ DATA.
AE17	2	Dirección del comienzo de lectura de DATA para RESTORE.
AE1B	2	Usado por ON ERROR GOTO.
AE1F	1	Bit: 0 = TROFF, 1 = TRON.
AE55	2	Guardado de DE para usarlo en la instrucción CALL.

LAS PRINCIPALES VARIABLES DE SISTEMA DEL CPC664

Dirección	*Longitud*	*Significado*
AE57	1	Guardado del acumulador para usarlo en la instrucción CALL.
AE58	2	Guardado de HL para usarlo en la instrucción CALL.
AE5A	2	Guardado de SP para usarlo en la instrucción CALL.
AE5C	2	Usado por la instrucción ZONE (dirección).
AE5E	2	HIMEM (dirección superior para el BASIC).
AE60	2	Usado por la instrucción SYMBOL (dirección).
AE64	2	Dirección del comienzo del programa BASIC. (Por omisión es 016F.)
AE66	2	Dirección de fin del programa BASIC.
AE68	2	Dirección de comienzo de las tablas de variables.
AE6A	2	Dirección de la tabla de variables simples.
AE6C	2	Dirección de la tabla de variables de tablas (DIM).
B06F	2	Dirección de comienzo de la pila BASIC.
B09F	1	Tipo del acumulador virtual.
B0A0	5	Cinco bytes usados por el acumulador virtual.
B113	1	Modo radianes/grados.
B118		Comienzo de la zona de las variables del gestor de cassette.
B118	1	Indicador PROMPT permitido si = 0, impedido si ≠ 0.
B11A	1	Indicador de apertura de fichero.
B11B	2	Dirección del buffer de 2K para el directorio.
B11D	2	Dirección del buffer de lectura.
B131	1	Tipo del fichero de cassette.
B132	2	Dirección actual del buffer de lectura.
B134	2	Ubicación de la memoria de datos.

LAS PRINCIPALES VARIABLES DE SISTEMA DEL CPC664

Dirección	*Longitud*	*Significado*
B136	2	Longitud lógica del fichero.
B15F	1	Tipo del stream de escritura.
B162	2	Dirección del buffer de escritura.
B176	2	Dirección actual del buffer de escritura.
B1E5	1	Carácter de sincronismo.
B1E9	2	Velocidad de lectura o escritura.
B1ED		Comienzo de la zona de variables del gestor sonoro.
B2A6	240	15 × 16 bytes con los valores de las envolventes de amplitud.
B396	240	15 × 16 bytes con los valores de las envolventes tonales.
B496	80	Tabla de los valores de las teclas sin SHIFT ni CTRL.
B4E6	80	Tabla de los valores de las teclas con SHIFT.
B536	80	Tabla de los valores de las teclas con CTRL.
B586	80	Tabla de repeticiones para cada tecla.
B628	2	Usado para el scanning (dirección).
B62A	1	Guardado temporal del carácter del scanning.
B633	1	Valor de la velocidad de repetición de las teclas.
B634	1	Valor del retardo antes de la repetición de una tecla.
B635	10	Tabla de scanning de las teclas.
B63B	1	Estado de la manecilla de juegos 1.
B63E	1	Estado de la manecilla de juegos 2.
B68B	2	Dirección de la tabla de teclas sin SHIFT ni CTRL.
B68D	2	Dirección de la tabla de teclas con SHIFT.
B68F	2	Dirección de la tabla de teclas con CTRL.

LAS PRINCIPALES VARIABLES DE SISTEMA DEL CPC664

Dirección	*Longitud*	*Significado*
B691	2	Dirección de la tabla de repetición de las teclas.
B693	2	Coordenada del origen del eje X.
B695	2	Coordenada del eje Y.
B697	2	Coordenada gráfica X.
B699	2	Coordenada gráfica Y.
B69B	2	Coordenada X de un borde de la ventana gráfica.
B69D	2	Coordenada X del otro borde de la ventana gráfica.
B69F	2	Coordenada Y de un borde de la ventana gráfica.
B6A1	2	Coordenada Y del otro borde de la ventana gráfica.
B6A3	1	Tinta de la pluma gráfica.
B6A4	1	Tinta del papel gráfico.
B6A5	8	Cuatro zonas de dos bytes utilizados como memoria temporal de cálculo durante el trazado de una línea.
B6AD	2	Coordenada X del punto final para el trazado de una línea.
B6AF	2	Coordenada Y del punto final para el trazado de una línea.
B6B5	1	Número del STREAM.
B726	1	Posición línea cursor.
B727	1	Posición columna cursor.
B728	1	Indicador de ventana.
B729	1	Línea de fin de la ventana actual.
B72A	1	Columna de fin de la ventana actual.
B72B	1	Línea de fin de la ventana actual.
B72C	1	Columna de fin de la ventana actual.
B72E	1	0 = cursor permitido, 255 = cursor no permitido.
B72F	1	Tinta actual para la pluma.

Dirección	*Longitud*	*Significado*
B730	1	Tinta actual para el papel.
B731	1	0 = permitida la visualización del fondo. 255 = no permitida la visualización del fondo.
B734	2	Primer carácter y estado de la tabla de matrices definida por el usuario.
B736	2	Dirección de la tabla de matrices definida por el usuario.
B763	96	Tabla de los códigos de control.
B7C2	1	Byte para la codificación (máscara) de la tinta.
B7C3	1	Modo de la pantalla (0, 1 o 2).
B7C4	2	OFFSET de la pantalla comprendido entre 0 y 7FF.
B7C6	1	Byte más significativo del inicio de la memoria real de pantalla.
B7C7	1	Contiene a veces un C3 (jump).
B7C8	2	Contiene la dirección para el jump.
B7D2	1	Duración del primer periodo de centelleo del marco.
B7D3	1	Duración del segundo periodo de centelleo del marco.
B7D4	32	Colores de las tintas (dos bytes por color).
B7F7	1	Usado por BORDER.

DIRECCIONES PRINCIPALES DE LA ROM INFERIOR DEL CPC664

La ROM inferior contiene las rutinas de sistema (comunicación con hardware), las rutinas matemáticas y el generador de caracteres.

Nota: las direcciones que corresponden a las rutinas ya descritas detalladamente se indican solamente con el punto de entrada a RAM que les corresponde seguido de un *.

Por tanto se aconseja consultar las páginas 81 a 110 para tener una mejor información.

Las rutinas situadas en una dirección idéntica, en el CPC664, se indican con el signo = colocado después de la dirección.

005C=	BCC8 *
0099=	BD0D *
00A3=	BD10 *
0163=	BCD7 *
016A=	BCDA *
0170=	BCDD *
0176=	BCE0 *
017D=	BCE3 *
0183=	BCE6 *
01B3=	BCE9 *
01C5=	BCEC *
01D2=	BCEF *
01E2=	BCF2 *
0219	BCFE *
0227	BCF5 *
0255	BCFB *
0276	BD01 *
0284	BCF8 *
028D	BD0A *
0294	BD04 *
029A	BD07 *
02A0	BCD1 *
02B1	BCD4 *
0326	BCCB *
0330	BCCE *
05D7	BD13 *
0606	BD16 *
066F	Mensaje : 64K MICROCOMPUTER (V2)
068B	Mensaje : copyright 1984 Amstrad Consumer Electronics PLC and Locomotive Software Ltd
06F5	Mensaje : *** program load failed ***
0728	Lista de compatibles Arnold, Amstrad, Orion, Schneider, Awa, Solavox, Saisho, Triumph, Isp
0766	BD1C *
0776	BD22 *
077C	BD25 *
07A4	BD19 *

07B0	BD1F *
07D0	BD28 *
080B	BD2B *
0825	BDF1 *
0834	BD31 *
0848	BD2E *
0853	BD34 *
08BB	BD37 *
0ABB	BBFF *
0ACC	BC02 *
0AE5	BC0E *
0B08	BC11 *
0B13	BC14 *
0B13	BDEB *
0B33	BC05 *
0B38	BC08 *
0B52	BC0B *
0B59	BC17 *
0B66	BC1A *
0BAB	BC1D *
0C01	BC20 *
0C0D	BC23 *
0C1B	BC26 *
0C35	BC29 *
0C51	BC59 *
0C6D	BDE8 *
0C70	BC5C *
0C86	BDE5 *
0C8A	BC2C *
0CA3	BC2F *
0CE6	BC3E *
0CEA	BC41 *
0CEE	BC32 *
0CF3	BC38 *
0D16	BC35 *
0D1B	BC3B *
0DB5	BC44 *
0DB9	BC47 *
0DE1	BC4A *
0DFC	BC4D *
0E40	BC50 *
0EF5	BC53 *
0F26	BC56 *
0F8F	BC5F *
0F97	BC62 *
1070	BB4E *
1080	BB51 *
10E0	BBB4 *
10FF	BBB7 *
1156	BB6F *
1161	BB72 *
116C	BB75 *
1178	BB78 *
11C6	BB87 *
1204	BB66 *
124E	BB69 *
125B	BDCD *
125B	BDD0 *
1261	BB8A *
1261	BB8D *
1272	BB81 *
127A	BB84 *
1282	BB7B *
1293	BB7E *
12A2	BB90 *
12A7	BB96 *
12B6	BB93 *
12BC	BB99 *
12C2	BB9C *
12D0	BBA5 *
12EE	BBA8 *
12FA	BBAB *
1327	BBAE *
1331	BB5D *
1347	BDD3 *
1377	BB9F *
1384	BBA2 *
13A4	BB63 *
13A8	BB60 *
13BA	BDD6 *
13FA	BB5A *
1406	BDD9 *
144E	BB57 *
1455	BB54 *
14D0	BBB1 *
154B	BB6C *
15A4	BBBA *
15D3	BBBD *
15F7	BBC3 *
15FA	BBC0 *
1602	BBC6 *
160A	BBC9 *
1618	BBCC *
16A1	BBCF *
16E6	BBD2 *
1713	BBD5 *

DIRECCIONES PRINCIPALES DE LA ROM INFERIOR DEL CPC664

1729	BBD8 *
1732	BBDB *
1763	BBDE *
176A	BBE4 *
1771	BBE1 *
1776	BBE7 *
177C	BBED *
177F	BBEA *
1782	BDDC *
1790	BBF3 *
1793	BBF0 *
1796	BDDF *
17A2	BBF9 *
17A5	BBF6 *
17B0	BDE2 *
193C	BBFC *
1B5C	BB00 *
1B98	BB03 *
1BBF	BB06 *
1BC5	BB09 *
1BFA	BB0C *
1C04	BB15 *
1C3C	Valor por omisión de las teclas extendidas (RUN para CTRL CR)
1C46	BB0F *
1CB3	BB12 *
1CDB	BB18 *
1CE1	BB1B *
1D38	BB21 *
1DB8	BDEE *
1DE5	BB24 *
1DF2	BB42 *
1DF6	BB3F *
1DFA	BB45 *
1E0B	BB48 *
1E19	BB4B *
1E2F	BB3C *
1E34	BB39 *
1E45	BB1E *
1EC4	BB2A *
1EC9	BB30 *
1ECE	BB36 *
1ED8	BB27 *
1EDD	BB2D *
1EE2	BB33 *
1EEF	Tabla de los valores por omisión de las teclas del teclado
1FE9	BCA7 *
2050	BCB6 *
206B	BCB9 *

2114	BCAA *	
21AC	BCB3 *	
21CE	BCAD *	
21EB	BCB0 *	
2495	BCBC *	
249A	BCBF *	
24A6	BCC2 *	
24AB	BCC5 *	
24BC	BC65 *	
24CE	BC68 *	
24E1	BC6B *	
288B	BC77, BC7A, BC7D, BC80, BC83, BC86, BC89, BC8C, BC8F, BC92, BC95, BC98, BC9B *	
	Rutinas cassette y disco	
2935	Mensaje :	press play then any key
294B	Mensaje :	error
2955	Mensaje :	REC
2958	Mensaje :	and
295D	Mensaje :	Read
2963	Mensaje :	write
296A	Mensaje :	Rewind
2970	Mensaje :	tape
2975	Mensaje :	found
297D	Mensaje :	loading
2985	Mensaje :	saving
298D	Mensaje :	OK
2990	Mensaje :	Block
2996	Mensaje :	Unnamed
299D	Mensaje :	File
29A6	BCA1 *	
29AF	BC9E *	
29C1	BCA4 *	
2BBB	BC6E *	
2BBF	BC71 *	
2BC1	BC74 *	
2F73	BD97 * PI	
2F78	CONSTANTE PI	
2F91	BD5E *	
2F9F	BD61 *	
2FC8	BD64 *	
2FD1	BDB5 *	
2FD9	BD67 *	
3001	BD6A *	
3014	BD6D *	
3055	BD70 *	
305F	BD73 *	
30C6	BD76 *	

DIRECCIONES PRINCIPALES DE LA ROM INFERIOR DEL CPC664

30F5	Tabla de las potencias de 10. Trece entradas de 5 bytes para los valores de 10 a 10E13
3136	BDB8 * RND INT
3143	BDBB * RND SEED
3159	BD7C * RND
3188	BD88 * RND
31B1	BDA3 * LOG10
31B6	BDA0 * LOG
31EE	Constante para el cálculo del LOG (4 × 5 bytes)
3220	Valor codificado de 1/SQR(2)
3225	Valor codificado de LOG(2) (0,693147181)
322A	Valor codificado de LOG10(2) (0,301029996)
322F	BDA6 * EXP
329D	Constante 1,44269504
32A2	Constante 88,0296919
32A7	Constante −88,7228391
32AC	BD9A * SQR
32AF	BD9C * POTENCIACIÓN
3345	BD94 * DEG-RAD
3349	BDAC * COS
3353	BDA9 * SIN
3382	Tabla de 6 números codificados en 5 bytes cada uno para el cálculo de los senos y cosenos
33B4	Tabla de 4 números codificados en 5 bytes cada uno para el cálculo de los senos y cosenos
33C8	BDAF * TAN
33D8	BDB2 * ATN
33EE	Tabla de 11 números codificados en 5 bytes cada uno para el cálculo del arcotangente
349E	BD7F * SUSTRACCIÓN
34A2	BD79 * ADICIÓN
3577	BD82 * MULTIPLICACIÓN
3604	BD85 * DIVISIÓN
36DF	BD8B * COMPARACIÓN
3727	BD91 * SGN
3731	BD8E * CAMBIO DE SIGNO
3800	Comienzo de la tabla del generador de caracteres (256 × 8 bytes)
3FFF	Fin de la tabla

DIRECCIONES PRINCIPALES DE LA ROM SUPERIOR DEL CPC664

La ROM superior contiene todas las rutinas de tratamiento de todas las palabras-clave del BASIC.

C006	Inicialización + envío del mensaje : BASIC 1.1
C033	Mensaje : BASIC 1.1
C046	Función EDIT
C058	Entrada principal (visualización de READY)
C0D7	Mensaje : READY
C0EA	AUTO
C128	NEW
C12F	CLEAR
C23C	PAPER
C227	PEN
C24B	BORDER
C254	INK
C278	MODE
C283	CLS
C29B	COPYCHR$
C2A4	VPOS
C2A8	POS
C302	LOCATE
C311	WINDOW
C346	TAG
C34D	TAGOFF
C363	CURSOR
C42D	WIDTH
C452	EOF
C4E1	ORIGIN
C509	CLG
C515	FILL
C532	MOVE
C537	MOVER
C53C	DRAW
C541	DRAWR
C546	PLOT
C54B	PLOTR
C574	TEST
C579	TESTR
C59D	GRAPHICS
C5C3	MASK
C5D7	FOR
C6A5	NEXT
C76A	IF
C789	GOTO
C78F	GOSUB
C7B3	RETURN
C7EA	WHILE
C81D	WEND

DIRECCIONES PRINCIPALES DE LA ROM SUPERIOR DEL CPC664

C885	ON
C979	ON BREAK
C99A	DI
C9A0	EI
C9F8	ON SQ
CA25	AFTER
CA2D	EVERY
CA53	REMAIN
CB54	ERROR
CBF4	Mensaje : UNDEFINED LINE
CC04	Rutina de envío del mensaje BREAK in
CC1F	Mensaje : BREAK
CC25	Mensaje : IN
CC29	STOP
CC34	END
CC96	CONT
CCCD	ON ERROR
CCD8	RESUME
CD17	Tabla de los mensajes de error (FRACCIONES DE PALABRA)
CFF0	Tabla de los puntos de entrada de las operaciones aritméticas y lógicas
D11A	Tabla de los puntos de entrada de las funciones EOF, ERR, HIMEM, INKEY$, PI, RND, TIME, XPOS e YPOS
D12E	DERR
D133	ERR
D14B	HIMEM
D164	XPOS
D16B	YPOS
D1E8	Tabla de los puntos de entrada de las funciones
D242	MIN
D246	MAX
D26D	ROUND
D2AB	OPENOUT
D2B7	OPENIN
D2F0	CLOSEIN
D2F8	CLOSEOUT
D316	SOUND
D373	RELEASE
D37E	SQ
D3A1	ENV
D3D7	ENT
D459	INKEY
D473	JOY
D489	KEY DEF
D4DE	SPEED
D520	PI
D52C	DEG
D530	RAD
D534	SQR

D539	Rutina de elevación a una potencia
D563	EXP
D568	LOG10
D56D	LOG
D572	SIN
D577	COS
D57C	TAN
D581	ATN
D587	Mensaje : RANDOM NUMBER SEED?
D59C	RANDOMIZE
D5C4	RND
D653	DEFSTR
D657	DEFINT
D65B	DEFREAL
D691	LET
D6B9	DIM
D9F4	ERASE
DB18	LINE
DB48	INPUT
DB7F	Mensaje : ?redo from start
DCCD	RESTORE
DCDF	READ
DEC6	TRON
DECA	TROFF
DEE5	Tabla de los puntos de entrada de las palabras-clave BASIC
DFA8	Fin de la tabla
E0C8	Tabla de las palabras-clave que pueden ir seguidas de un número de línea (GOTO, RESTORE, AUTO, EDIT...)
E1D2	LIST
E3AD	Rutina de posicionamiento en la tabla de las letras para buscar las palabras-clave
E3F0	Rutina de comprobación que verifica si una palabra-clave se halla en la tabla
E41D	Tabla de las direcciones para cada una de las 26 letras del alfabeto
E451	Tabla de las palabras-clave con su código
E73A	Fin de la tabla
E7F3	DELETE
E8A3	RENUM
E9A8	DATA
E9AC	REM
EA7D	RUN
EABA	LOAD
EB02	CHAIN
EB59	MERGE
ECE1	SAVE
F20D	PEEK
F214	POKE
F21E	INP
F228	OUT

DIRECCIONES PRINCIPALES DE LA ROM SUPERIOR DEL CPC664

F232	WAIT
F261	CALL
F2A2	ZONE
F2A9	PRINT
F383	PRINT USING
F50D	WRITE
F570	MEMORY
F784	SYMBOL
F8EC	LOWER$
F8F1	Rutina de conversión a minúsculas
F8FA	UPPER$
F964	BIN$
F969	HEX$
F98F	DEC$
F9BC	STR$
F9D3	LEFT$
F9D8	RIGHT$
FA07	MID$
FA69	LEN
FA6E	ASC
FA74	CHR$
FA7E	INKEY$
FA8D	STRING$
FAAD	SPACE$
FABE	VAL
FAE5	INSTR
FC53	FRE
FD0C	Adición +
FD21	Sustracción –
FD35	Multiplicación *
FD52	División /
FD67	División entera \
FD79	Módulo (resto de la división)
FD87	Función AND (Y LÓGICO)
FD92	Función OR (O LÓGICO)
FD9C	Función XOR (O EXCLUSIVO)
FDB0	ABS
FE0E	FIX
FE13	INT
FEB6	CINT
FEEB	UNT
FF14	CREAL
FF1B	Vaciado del acumulador
FF2A	SGN
FF32	Posicionamiento de un entero en el acumulador
FF3E	Conversión a número real
FF45	Mete el tipo de variable en C
FF4B	Mete el tipo de variable en A

FF83	Copia el acumulador en la zona apuntada por DE
FF92	Comprueba si es mayúscula
FF9C	Comprueba si es numérica
FFAB	Conversión a mayúscula
FFCA	Compara A y el contenido de HL
FFD8	Compara HL y DE
FFDE	Compara HL y BC
FFE4	DE = HL − DE
FFF2	LDIR
FFF8	LDDR
FFFB	JP (HL)
FFFC	Retorno al contenido de BC
FFFE	Retorno al contenido de DE

DIRECCIONES REALES ROM DEL CPC664

Dirección vector	*Dirección real*	*Dirección vector*	*Dirección real*	*Dirección vector*	*Dirección real*
BBØØ	1B5C	BBØ3	1B98	BBØ6	1BBF
BBØ9	1BC5	BBØC	1BFA	BBØF	1C46
BB12	1CB3	BB15	1CØ4	BB18	1CDB
BB1B	1CE1	BB1E	1E45	BB21	1D38
BB24	1DE5	BB27	1ED8	BB2A	1EC4
BB2D	1EDD	BB3Ø	1EC9	BB33	1EE2
BB36	1ECE	BB39	1E34	BB3C	1E2F
BB3F	1DF6	BB42	1DF2	BB45	1DFA
BB48	1EØB	BB4B	1E19	BB4E	1Ø7Ø
BB51	1Ø8Ø	BB54	1455	BB57	144E
BB5A	13FA	BB5D	1331	BB6Ø	13A8
BB63	13A4	BB66	12Ø4	BB69	124E
BB6C	154B	BB6F	1156	BB72	1161
BB75	116C	BB78	1178	BB7B	1282
BB7E	1293	BB81	1272	BB84	127A
BB87	11C6	BB8A	1261	BB8D	1261
BB9Ø	12A2	BB93	12B6	BB96	12A7
BB99	12BC	BB9C	12C2	BB9F	1377
BBA2	1384	BBA5	12D0	BBA8	12EE
BBAB	12FA	BBAE	1327	BBB1	14D0
BBB4	1ØEØ	BBB7	1ØFF	BBBA	15A4
BBBD	15D3	BBCØ	15FA	BBC3	15F7
BBC6	16Ø2	BBC9	16ØA	BBCC	1618
BBCF	16A1	BBD2	16E6	BBD5	1713
BBD8	1729	BBDB	1732	BBDE	1763
BBE1	1771	BBE4	176A	BBE7	1776
BBEA	177F	BBED	177C	BBFØ	1793
BBF3	179Ø	BBF6	17A5	BBF9	17A2
BBFC	193C	BBFF	ØABB	BCØ2	ØACC
BCØ5	ØB33	BCØ8	ØB38	BCØB	ØB52
BCØE	ØAE5	BC11	ØBØ8	BC14	ØB13
BC17	ØB59	BC1A	ØB66	BC1D	ØBAB
BC2Ø	ØCØ1	BC23	ØCØD	BC26	ØC1B
BC29	ØC35	BC2C	ØC8A	BC2F	ØCA3
BC32	ØCEE	BC35	ØD16	BC38	ØCF3
BC3B	ØD1B	BC3E	ØCE6	BC41	ØCEA
BC44	ØDB5	BC47	ØDB9	BC4A	ØDE1
BC4D	ØDFC	BC5Ø	ØE4Ø	BC53	ØEF5
BC56	ØF26	BC59	ØC51	BC5C	ØC7Ø
BC5F	ØF8F	BC62	ØF97	BC65	24BC
BC68	24CE	BC6B	24E1	BC6E	2BBB
BC71	2BBF	BC74	2BC1	BC77	288B

DIRECCIONES REALES ROM DEL CPC664

Dirección vector	*Dirección real*	*Dirección vector*	*Dirección real*	*Dirección vector*	*Dirección real*
BC7A	288B	BC7D	288B	BC8Ø	288B
BC83	288B	BC86	288B	BC89	288B
BC8C	288B	BC8F	288B	BC92	288B
BC95	288B	BC98	288B	BC9B	288B
BC9E	29AF	BCA1	29A6	BCA4	29C1
BCA7	1FE9	BCAA	2114	BCAD	21CE
BCBØ	21EB	BCB3	21AC	BCB6	2Ø5Ø
BCB9	2Ø6B	BCBC	2495	BCBF	249A
BCC2	24A6	BCC5	24AB	BCC8	ØØ5C
BCCB	Ø326	BCCE	Ø33Ø	BCD1	Ø2AØ
BCD4	Ø2B1	BCD7	Ø163	BCDA	Ø16A
BCDD	Ø17Ø	BCEØ	Ø176	BCE3	Ø17D
BCE6	Ø183	BCE9	Ø1B3	BCEC	Ø1C5
BCEF	Ø1D2	BCF2	Ø1E2	BCF5	Ø227
BCF8	Ø284	BCFB	Ø255	BCFE	Ø219
BDØ1	Ø276	BDØ4	Ø294	BDØ7	Ø29A
BDØA	Ø28D	BDØD	ØØ99	BD1Ø	ØØA3
BD13	Ø5D7	BD16	Ø6Ø6	BD19	Ø7A4
BD1C	Ø766	BD1F	Ø7BØ	BD22	Ø776
BD25	Ø77C	BD28	Ø7DØ	BD2B	Ø8ØB
BD2E	Ø848	BD31	Ø834	BD34	Ø853
BD37	Ø8BB	BD3A	1D3C	BD3D	1BFE
BD4Ø	145C	BD43	15E8	BD46	19D1
BD49	17AC	BD4C	17A8	BD4F	1626
BD52	19D5	BD55	ØB41	BD58	Ø7FC
BD5B	2CØ2	BD5E	2F91	BD61	2F9F
BD64	2FC8	BD67	2FD9	BD6A	3ØØ1
BD6D	3Ø14	BD7Ø	3Ø55	BD73	3Ø5F
BD76	3ØC6	BD79	34A2	BD7C	3159
BD7F	349E	BD82	3577	BD85	36Ø4
BD88	3188	BD8B	36DF	BD8E	3731
BD91	3727	BD94	3345	BD97	2F73
BD9A	32AC	BD9D	32AF	BDAØ	31B6
BDA3	31B1	BDA6	322F	BDA9	3353
BDAC	3349	BDAF	33C8	BDB2	33D8
BDB5	2FD1	BDB8	3136	BDBB	3143

DIRECCIONES DE EJECUCIÓN DE LAS PALABRAS-CLAVE DEL BASIC DEL CPC664

Palabra-clave	*Dirección*	*Palabra-clave*	*Dirección*
ABS	FDBØ	FRE	FC53
AFTER	CA25	GOSUB	C78F
ASC	FA6E	GOTO	C789
ATN	D581	HEX$	F969
AUTO	CØEA	HIMEM	D14B
BIN$	F964	IF	C76A
BORDER	C24B	INSTR	FAE5
CALL	F261	INK	C254
CAT	D299	INKEY	D459
CHAIN	EBØ2	INKEY$	FA7E
CHR$	FA74	INP	F21E
CINT	FEB6	INPUT	DB48
CLEAR	C12F	INT	FE13
CLG	C5Ø9	JOY	D473
CLOSEIN	D2FØ	KEY	D489
CLOSEOUT	D2F8	LEFT$	F9D3
CLS	C283	LEN	FA69
CONT	CC96	LET	D691
COS	D577	LINE	DB18
CREAL	FF14	LIST	E1D2
DATA	E9A8	LOAD	EABA
DEC$	F9F8	LOCATE	C302
DEF	D174	LOG	D56D
DEFINT	D657	LOG10	D568
DEFREAL	D65B	LOWER$	F8EC
DEFSTR	D653	MAX	D246
DEG	D52C	MEMORY	F570
DELETE	E7F3	MERGE	EB59
DI	C99A	MID$	FAØ7
DIM	D6B9	MIN	D242
DRAW	C53C	MODE	C278
DRAWR	C541	MOVE	C532
EDIT	CØ46	MOVER	C537
EI	C9AØ	NEXT	C6A5
ELSE	E9B2	NEW	C128
END	CC34	ON	C885
ENT	D3D7	ON BREAK	C979
ENV	D3A1	ON ERROR	CCCD
EOF	C452	ON SQ	C9F8
ERASE	D9F4	OPENIN	D2B7
ERR	D133	OPENOUT	D2AB
ERROR	CB54	ORIGIN	C4E1
EVERY	CA2D	OUT	F228
EXP	D563	PAPER	C23C
FIX	FEØE	PEEK	F2QD
FOR	C5D7	PEN	C227

DIRECCIONES DE EJECUCIÓN DE LAS PALABRAS-CLAVE DEL BASIC DEL CPC664

Palabra-clave	*Dirección*	*Palabra-clave*	*Dirección*
PI	D520	SQ	D37E
PLOT	C546	SQR	D534
PLOTR	C54B	STOP	CC29
POKE	F214	STR$	F9CB
POS	C2AD	STRING$	FA8D
PRINT	F2A9	SYMBOL	F784
' (REM)	E9AC	TAG	C346
RAD	D530	TAGOFF	C34D
RANDOMIZE	D59C	TAN	D57C
READ	DCDF	TEST	C574
RELEASE	D373	TESTR	C579
REM	E9AC	TIME	D13C
REMAIN	CA53	TROFF	DEC6
RENUM	E8A3	TRON	DECA
RESTORE	DCCD	UNT	FEEB
RESUME	CCD8	UPPER$	F8FA
RETURN	C7B3	VAL	FABE
RIGHT$	F9D8	VPOS	C2A4
RND	D5C4	WAIT	F2E2
ROUND	D26D	WEND	C81D
RUN	EA7D	WHILE	C7EA
SAVE	ECE1	WIDTH	C42D
SGN	FF2A	WINDOW	C311
SIN	D572	WRITE	F50D
SOUND	D316	XPOS	D164
SPACE$	FAAD	YPOS	D16B
SPEED	D4DE	ZONE	F2A2

Nuevas palabras-clave:

COPYCHR$	C29B	FRAME	BD19
CURSOR	C363	GRAPHICS	C59D
DERR	D12E	MASK	C5C3
FILL	C515		

La versión CPC6128 ofrece algunas diferencias con respecto a la versión CPC664.

La ROM inferior está algo modificada. La ROM superior es casi idéntica, pero todas las direcciones están ligeramente desplazadas.

No obstante, los vectores de sistema, las variables de sistema y las rutinas matemáticas son exactamente idénticas a las del CPC664 en cuanto a la función y dirección.

Las páginas siguientes ofrecen las modificaciones sufridas por las ROMs, BIOS y BASIC.

DIRECCIONES PRINCIPALES DE LA ROM INFERIOR DEL CPC6128

La ROM inferior contiene las rutinas de sistema (comunicación con el material), las rutinas matemáticas y el generador de caracteres.

Nota: las direcciones que corresponden a las rutinas ya descritas detalladamente se indican solamente con el punto de entrada en RAM que le corresponde seguido de un *.

Por tanto es aconsejable consultar las páginas 81 a 110 para tener una mejor información.

Las rutinas situadas en idéntica dirección que en el CPC664 se indican con el signo = colocado después de la dirección.

005C=	BCC8 *
0099=	BD0D *
00A3=	BD10 *
0163=	BCD7 *
016A=	BCDA *
0170=	BCDD *
0176=	BCE0 *
017D=	BCE3 *
0183=	BCE6 *
01B3=	BCE9 *
01C5=	BCEC *
01D2=	BCEF *
01E2=	BCF2 *
0219=	BCFE *
0227=	BCF5 *
0255=	BCFB *
0276=	BD01 *
0284=	BCF8 *
028D=	BD0A *
0294=	BD04 *
029A=	BD07 *
02A0=	BCD1 *
02B1=	BCD4 *
0326=	BCCB *
0330=	BCCE *
05ED	BD13 *
061C	BD16 *
0688	Mensaje : 128K MICROCOMPUTER (V3)
068B	Mensaje : copyright 1985 Amstrad Consumer Electronics PLC and Locomotive Software Ltd
06F5	Mensaje : *** program load failed ***
0728	Lista de compatibles Arnold, Amstrad, Orion, Schneider, Awa, Solavox, Saisho, Triumph, Isp
0776	BD1C *
0786	BD22 *
078C	BD25 *
07B4	BD19 *
07C0	BD1F *

07E0	BD28 *
081B	BD2B *
0835	BDF1 *
0844	BD31 *
0858	BD2E *
0863	BD34 *
08BD	BD37 *
0ABF	BBFF *
0AD0	BC02 *
0AE9	BC0E *
0B0C	BC11 *
0B17	BC14 *
0B17	BDEB *
0B37	BC05 *
0B3C	BC08 *
0B56	BC0B *
0B5D	BC17 *
0B6A	BC1A *
0BAF	BC1D *
0C05	BC20 *
0C11	BC23 *
0C1F	BC26 *
0C39	BC29 *
0C55	BC59 *
0C71	BDE8 *
0C74	BC5C *
0C8A	BDE5 *
0C8E	BC2C *
0CA7	BC2F *
0CEA	BC3E *
0CEE	BC41 *
0CF2	BC32 *
0CF7	BC38 *
0D1A	BC35 *
0D1F	BC3B *
0DB9	BC44 *
0DBD	BC47 *
0DE5	BC4A *
0E00	BC4D *
0E44	BC50 *
0EF9	BC53 *
0F2A	BC56 *
0F93	BC5F *
0F9B	BC62 *
1074	BB4E *
1084	BB51 *
10E4	BBB4 *
1103	BBB7 *
115A	BB6F *
1165	BB72 *
1170	BB75 *
117C	BB78 *
11CA	BB87 *
1208	BB66 *
1252	BB69 *
125F	BDCD *
125F	BDD0 *
1265	BB8A *
1265	BB8D *
1276	BB81 *
127E	BB84 *
1286	BB7B *
1297	BB7E *
12A6	BB90 *
12AB	BB96 *
12BA	BB93 *
12C0	BB99 *
12C6	BB9C *
12D4	BBA5 *
12E2	BBA8 *
12FE	BBAB *
132B	BBAE *
1335	BB5D *
134B	BDD3 *
137B	BB9F *
1388	BBA2 *
13A8	BB63 *
13AC	BB60 *
13BE	BDD6 *
13FE	BB5A *
140A	BDD9 *
1452	BB57 *
1459	BB54 *
14D4	BBB1 *
154F	BB6C *
15A8	BBBA *
15D7	BBBD *
15FB	BBC3 *
15FE	BBC0 *
1606	BBC6 *
160E	BBC9 *
161C	BBCC *
16A5	BBCF *
16EA	BBD2 *
1717	BBD5 *
172D	BBD8 *

DIRECCIONES PRINCIPALES DE LA ROM INFERIOR DEL CPC6128

1736	BBDB *
1767	BBDE *
176E	BBE4 *
1775	BBE1 *
177A	BBE7 *
1780	BBED *
1783	BBEA *
1786	BDDC *
1794	BBF3 *
1797	BBF0 *
179A	BDDF *
17A6	BBF9 *
17A9	BBF6 *
17B4	BDE2 *
1940	BBFC *

1B5C= A partir de esta dirección, las rutinas de la ROM inferior del CPC6128 empiezan en los mismos puntos de entrada que las de la ROM inferior del CPC664.

DIRECCIONES PRINCIPALES DE LA ROM SUPERIOR DEL CPC6128

La ROM superior contiene todas las rutinas de tratamiento de todas las palabras-clave del BASIC.

C006	Inicialización + envío del mensaje : BASIC 1.1
C033	Mensaje : BASIC 1.1
C046	Función EDIT
C058	Entrada principal (visualización de READY)
C0D7	Mensaje : READY
C0EA	AUTO
C128	NEW
C12F	CLEAR
C224	PEN
C239	PAPER
C248	BORDER
C251	INK
C275	MODE
C280	CLS
C298	COPYCHR$
C2A1	VPOS
C2A5	POS
C2FF	LOCATE
C30E	WINDOW
C343	TAG
C34A	TAGOFF
C360	CURSOR
C42A	WIDTH
C44F	EOF
C4DE	ORIGIN
C506	CLG
C512	FILL
C52F	MOVE
C534	MOVER
C539	DRAW
C53E	DRAWR
C543	PLOT
C548	PLOTR
C571	TEST
C576	TESTR
C59A	GRAPHICS
C5C0	MASK
C5D4	FOR
C6A2	NEXT
C767	IF
C786	GOTO
C78C	GOSUB
C7B0	RETURN
C7E7	WHILE
C81A	WEND
C882	ON

DIRECCIONES PRINCIPALES DE LA ROM SUPERIOR DEL CPC6128

C976	ON BREAK
C997	DI
C99D	EI
C9F5	ON SQ
CA22	AFTER
CA2A	EVERY
CA50	REMAIN
CB51	ERROR
CBF1	Mensaje : UNDEFINED LINE
CC01	Rutina de envío del mensaje BREAK in
CC1C	Mensaje : BREAK
CC22	Mensaje : IN
CC26	STOP
CC31	END
CC93	CONT
CCCA	ON ERROR
CCD5	RESUME
CD14	Tabla de los mensajes de error (FRACCIONES DE PALABRA)
CFED	Tabla de los puntos de entrada de las operaciones aritméticas y lógicas
D01D	–
D028	NOT
D036	+
D117	Tabla de los puntos de entrada de las funciones EOF, ERR, HIMEM, INKEY$, PI, RND, TIME, XPOS e YPOS
D12B	DERR
D130	ERR
D139	TIME
D142	ERL
D148	HIMEM
D14E	@
D161	XPOS
D168	YPOS
D171	DEF
D1E5	Tabla de los puntos de entrada de las funciones
D23F	MIN
D243	MAX
DE6A	ROUND
D296	CAT
D2A8	OPENOUT
D2B4	OPENIN
D2ED	CLOSEIN
D2F5	CLOSEOUT
D313	SOUND
D370	RELEASE
D37B	SQ
D39E	ENV
D3D4	ENT
D456	INKEY

D470	JOY
D486	KEY DEF
D4DB	SPEED
D51D	PI
D529	DEG
D52D	RAD
D531	SQR
D536	Rutina de elevación a una potencia
D560	EXP
D565	LOG10
D56A	LOG
D56F	SIN
D574	COS
D579	TAN
D57E	ATN
D584	Mensaje : RANDOM NUMBER SEED?
D599	RANDOMIZE
D5C1	RND
D650	DEFSTR
D654	DEFINT
D658	DEFREAL
D68E	LET
D6B6	DIM
D9F0	ERASE
DB13	LINE
DB43	INPUT
DB7A	Mensaje : ? redo from start
DCC8	RESTORE
DCDA	READ
DEC1	TRON
DEC5	TROFF
DEE0	Tabla de los puntos de entrada de las palabras-clave BASIC
DFA3	Fin de la tabla
E0C3	Tabla de las palabras-clave que pueden ir seguidas de un número de línea (GOTO, RESTORE, AUTO, EDIT...)
E1CD	LIST
E3A8	Rutina de posicionamiento en la tabla de las letras para buscar las palabras-clave
E3EB	Rutina de comprobación que verifica si una palabra-clave se halla en la tabla
E418	Tabla de las direcciones para cada una de las 26 letras del alfabeto
E44C	Tabla de las palabras-clave con su código
E735	Fin de la tabla
E7EE	DELETE
E89E	RENUM
E9A3	DATA
E9A7	REM
E9AD	ELSE
EA78	RUN

DIRECCIONES PRINCIPALES DE LA
ROM SUPERIOR DEL CPC6128

EAB5	LOAD
EAFD	CHAIN
EB54	MERGE
ECDC	SAVE
F208	PEEK
F20F	POKE
F219	INP
F223	OUT
F229	WAIT
F25C	CALL
F29D	ZONE
F2A9	PRINT
F383	PRINT USING
F508	WRITE
F56B	MEMORY
F784	SYMBOL
F8EC	LOWER$
F8F1	Rutina de conversión a minúsculas
F8FA	UPPER$
F964	BIN$
F969	HEX$
F98F	DEC$
F9BC	STR$
F9D3	LEFT$
F9D8	RIGHT$
FA07	MID$
FA69	LEN
FA6E	ASC
FA74	CHR$
FA7E	INKEY$
FA8D	STRING$
FAAD	SPACE$
FABE	VAL
FAE5	INSTR
FC53	FRE
FD0C	Adición +
FD21	Sustracción –
FD35	Multiplicación *
FD52	División /
FD67	División entera
FD79	Módulo (resto de la división)
FD87	Función AND (Y LÓGICO)
FD92	Función OR (O LÓGICO)
FD9C	Función XOR (O EXCLUSIVO)
FDB0	ABS
FE0E	FIX
FE13	INT

FEB6	CINT
FEEB	UNT
FF14	CREAL
FF1B	Vaciado del acumulador
FF2A	SGN
FF32	Posicionamiento de un entero en el acumulador
FF3E	Conversión a número real
FF45	Mete el tipo de variable en C
FF4B	Mete el tipo de variable en A
FF83	Copia el acumulador en la zona apuntada por DE
FF92	Comprueba si es mayúscula
FF9C	Comprueba si es numérica
FFAB	Conversión a mayúscula
FFCA	Compara A y el contenido de HL
FFD8	Compara HL y DE
FFDE	Compara HL y BC
FFE4	DE = HL − DE
FFF2	LDIR
FFF8	LDDR
FFFB	JP (HL)
FFFC	Retorno al contenido de BC
FFFE	Retorno al contenido de DE

TABLA DE VALORES
PARA LA GAMA CROMÁTICA

DO	3822	FA♯	2703
DO♯	3608	SOL	2551
RE	3405	SOL♯	2408
RE♯	3214	LA	2273
MI	3034	LA♯	2145
FA	2863	SI	2025

Estos valores corresponden a la octava −3; para cada octava superior, basta con dividir los valores por 2.

TABLA DE CÓDIGOS DE CONTROL DEL TERMINAL

Código	*Acción*	*Parámetro*
0	--- --- --- ---	—
1	Imprime el carácter cuyo código sigue.	1
2	Suprime la visualización del cursor.	0
3	Permite la visualización del cursor.	0
4	Posiciona la pantalla en modo 0, 1, 2.	1
5	Impresión en modo gráfico del siguiente carácter.	1
6	Autorización de la visualización video.	0
7	Sonería.	0
8	Desplaza el cursor un carácter hacia atrás con borrado (BS).	0
9	Desplaza el cursor un carácter a la derecha.	0
10	Desplaza el cursor una línea hacia abajo.	0
11	Desplaza el cursor una línea hacia arriba.	0
12	Borrado de la ventana actual + HOME.	0
13	Retorno del carro.	0
14	Posiciona la tinta del papel.	1
15	Posiciona la tinta de la pluma.	1
16	Borrado del carácter actual.	0
17	Borrado del comienzo izquierdo de la ventana hasta la posición actual del cursor.	0
18	Borrado desde la posición actual del cursor hasta el borde derecho de la ventana.	0
19	Borrado del borde superior izquierdo de la ventana hasta la posición actual del cursor.	0
20	Borrado desde la posición actual del cursor hasta el borde inferior derecho de la ventana.	0
21	Inhibe la visualización.	
22	Posiciona modo opaco (0) o transparente (1).	1
23	Posiciona modo escritura gráfica.	1
24	Intercambia las tintas del papel y de la pluma.	0
25	Determina las matrices de un carácter.	9
26	Posiciona los límites de la ventana.	4
27	--- --- --- ---	—
28	Posiciona los colores de la tinta.	3
29	Posiciona los colores del borde.	2
30	Posiciona el cursor arriba a la izquierda de la ventana (HOME).	0
31	Posicionamiento absoluto del cursor en la ventana.	2

TABLA DE DIRECCIONES DE PUERTAS ÚTILES

7FXX	VIDEO GATE ARRAY	S
BCXX	6845 (DIRECCIÓN)	S
BDXX	6845 DATOS	S
BEXX	6845 STATUS (ESTADO)	E
BFXX	6845 DATOS	E
DFXX	SELECCIÓN NO EXTERNA	S
EFXX	PUERTA IMPRESORA	S
F4XX	8255 PUERTA A	E/S
F5XX	8255 PUERTA B	E/S
F6XX	8255 PUERTA C	E/S
	8255 PUERTA CONTROLADA	—
FFXX	RESERVADO AL USUARIO	

ESTRUCTURA DE LA MEMORIA DE PANTALLA

Tamaño : 16 K

Comienzo estándar : en C000, pero puede empezar en 0000, 4000 u 8000.

Sea cual sea el modo, la memoria de pantalla puede considerarse como 8000 palabras de 16 bits que definen 4, 8 o 16 puntos en los modos 0, 1 y 2, respectivamente.

MODO 0	4 puntos en 16 bits	4 bits por punto	16 colores
MODO 1	8 puntos en 16 bits	2 bits por punto	4 colores
MODO 2	16 puntos en 16 bits	1 bit por punto	1 color

Las líneas 0, 8, 16, 24..., 192 están codificadas en los dos primeros K.

Las líneas 1, 9, 17, 25..., 193 están codificadas en los dos siguientes K.

...

Las líneas 7, 15, 23, 31..., 199 están codificadas en los dos últimos K.

El registro de dirección 6845 determina la dirección de partida en el bloque de dos K (10 bits).

Cada línea utiliza 80 bytes consecutivos en memoria.

Por ejemplo, si la dirección de partida es C000, la línea 0 ocupa los 80 primeros bytes, de C000 a C04F; la línea 1 ocupa los 80 bytes de C800 a C84F, y la línea 8 ocupa los bytes C050 a C09F.

	Modo 0	*Modo 1*	*Modo 2*
Punto más a la izquierda	bits 1, 5, 3, 7	bits 3, 7	bit 7
...			bit 6
...		bits 2, 6	bit 5
...			bit 4
...	bits 0, 4, 2, 6	bits 1, 5	bit 3
...			bit 2
...		bits 0, 4	bit 1
Punto más a la derecha			bit 0

80 BYTES

200 LÍNEAS DE PUNTOS

C000	C001	....	C04E	C04F
C800	C801	....	C84E	C84F
D000	D001	....	D04E	D04F
D800	D801	....	D84E	D84F
F000	F001	....	F04E	F04F
F800	F801	....	F84E	F84F
C050	C051	....	C09E	C09F
C850	C851	....	C89E	C89F
:	:	....		
FF30	FF31	....	FF7E	FF7F
C780	C781	....	C7CE	C7CF
CF80	CF81	....	CFCE	CFCF
:	:	....		
F780	F781	....	F7CE	F7CF
FF80	FF81	....	FFCE	FFCF

LÍNEAS DEL PRIMER CARÁCTER

LÍNEAS DEL CARÁCTER 25

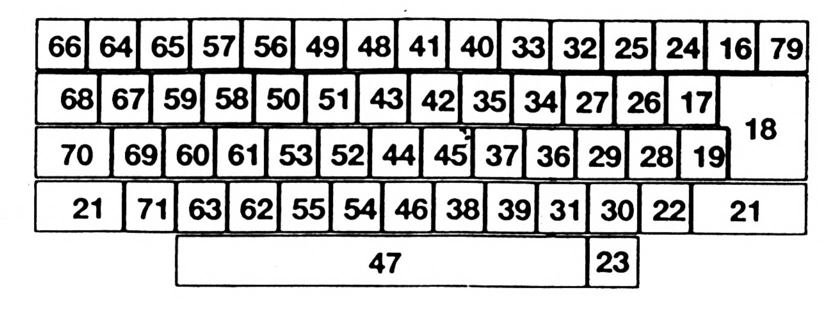
C7D0 ... C7FF, CFD0 ... CFFF, ..., FFD0 ... FFFF no se utilizan.

TABLA DE LOS COLORES

Número	*Color*	*Valor registro 6845*
0	Negro	20
1	Azul	4
2	Azul brillante	21
3	Rojo	28
4	Magenta	24
5	Malva	29
6	Rojo brillante	12
7	Violeta	5
8	Magenta brillante	13
9	Verde	22
10	Cian	6
11	Azul celeste	23
12	Amarillo	30
13	Blanco	0
14	Azul pastel	31
15	Naranja	14
16	Rosa	7
17	Magenta pastel	15
18	Verde brillante	18
19	Verde marino	2
20	Cian brillante	19
21	Amarillo limón	26
22	Verde pastel	25
23	Cian pastel	27
24	Amarillo brillante	10
25	Amarillo pastel	3
26	Blanco brillante	11

TABLA DE LOS CÓDIGOS DEL TECLADO (NÚMEROS DE LAS TECLAS)

Teclado



Teclado numérico

10	11	3
20	12	4
13	14	5
15	7	6

Flechas cursor

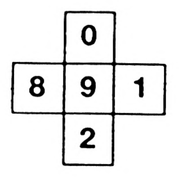


TABLA DE LOS CÓDIGOS DEL TECLADO (NÚMEROS DE LAS TECLAS)

Manecilla de juegos 0

	72	
74		75
	73	

Botón de tiro 1

77

Botón de tiro 2

76

Manecilla de juegos 1

	48	
50		51
	49	

Botón de tiro 1

53

Botón de tiro 2

52

ÍNDICE

memento

«CLAVES PARA EL AMSTRAD» es el manual que se abre en la página adecuada y permite acceder eficazmente a todas las informaciones que se necesitan: juego de instrucciones del Z80, puntos de entrada a las rutinas de sistema, bloques de control, estructura interna, programación, conectores y conexiones de los principales circuitos utilizados, etc.

«CLAVES PARA EL AMSTRAD» es también una pequeña colección de trucos: proteger un programa, producir sonidos especiales, hacer el scanning del teclado o introducir una rutina en lenguaje máquina son otros tantos truquillos para desvelar todas las posibilidades originales de su CPC 464, 664 o 6128.

No olvides las claves!!

CLAVES PARA AMSTRAD

CPC 464-664 y 6128 **Sistema de base**

ELISA



ISBN: 84-7622-021-9

EDICIONES ELISA S.A.

CLAVES PARA AMSTRAD

Daniel Martin

ELISA


AMSTRAD
CPC
OCR
MÉMOIRE ÉCRITE
MEMORY ENGRAVED
MEMORIA ESCRITA
ACME
https://acpc.me/